Installation & User Guide for LinuxPPC 1-9-9-9

Linux for PowerPC™ Systems, Release 5

LinuxPPC Linux for PowerPC™ systems

14 July, 1999

http://www.linuxppc.org

Table of Contents

***Credits**:*

Information on this guide is based on excellent documentation for Linux-pmac installation by Paul Mackerras, Jeff Carr's original PowerPC Linux Install and User Guide, Nick Bastin's PBG3 FAQ, Craig Sadler's G3 FAQ, Jon Howell's FAQ-O-Matic, William R Sowerbutts' BeBox Linux page, Chuck Schotten's BeBox Linux Scripts page, Geert Uytterhoeven's CHRP page, Cort Dougan's PReP page, as well as other tidbits gathered around the web, various read-me's and man files, mailing lists, newsgroups, and finally my own experiences. A big thanks to all the people who have e-mailed contributions and suggestions.

This guide was written to benefit the PowerPC Linux community as a whole, and dedicated to all the LinuxPPC developers worldwide.

Comments, contributions, and error reports regarding this guide can be sent to me at wj@linuxppc.org
If you have questions, please refer to the FAQ-O-Matic (http://www.dartmouth.edu/cgi-bin/cgiwrap/jonh/lppc/faq.pl) and the Linux-pmac mailing list and its archives (http://www.linuxppc.org/archive).

About LinuxPPC

LinuxPPC is a monolithic port of the Linux operating system, a free version of Unix originally created by Linus Torvalds that runs on a variety of computers (including x86, DEC Alphas, Sun Sparcs, M68000, MIPS, ARM). The source tree of LinuxPPC has been combined with the source tree of PowerMac/Linux (Linux-pmac), so the names Linux-pmac and LinuxPPC are often used interchangeably. LinuxPPC is worked on by people around the world and is led by the people at LinuxPPC. It is developed under the GNU General Public License (GPL) and its source code is freely available. LinuxPPC is completely compatible with programs compiled for MkLinux, except for the few that make Mach system calls. MkLinux is Apple Computer's version of Linux that runs on top of the Mach kernel.

LinuxPPC is a completely PowerPC-native operating system that can co-exist peacefully on the same hard drive as your original operating system, such as Mac OS, AIX, or BeOS. By placing LinuxPPC on separate partitions from your other operating system, it can function independently without interfering with any of your other data. You can also share files between the two systems by mounting the partitions within LinuxPPC. If you prefer, you can also install LinuxPPC on a separate hard drive or have it completely replace your original OS as your primary system.

A few features of LinuxPPC include:
- stable and fast - protected memory, preemptive multitasking, advanced virtual memory
- supports SCSI and IDE drives
- supports PCI-based 601, 603/603e, 604/604e/604r, and G3 machines (NuBus-based PowerMac 6100, 7100, and 8100 are not supported)
- shared library support, glibc 2.1, Linux 2.2 kernel
- compatible with both big and little endian filesystems
- floppy, sound, ethernet, and CD-ROM support
- serial and USB device support (can print to supported printers and use PPP through a modem).
- runs Netscape Communicator, MP3 players, and other popular internet/multimedia tools
- XWindows and several window managers (AfterStep, WindowMaker, Enlightenment, and desktop environments such as KDE and GNOME)
- free, powerful web server (Apache), anonymous FTP, file server, multi-user support
- Java support (JDK 1.1.7, 1.2)
- For Powermac users, BootX provides a simple OS chooser that allows you to select Mac OS or Linux at boot-up. You can also quit Mac OS and boot LinuxPPC from the Finder. Finally, BootX allows you to use Mac OS initialized video and install LinuxPPC without using a floppy
- X-based installer or Red Hat style installer

In addition, people are working on many other projects including emulators of the Mac OS that run within Linux. These include a port of SheepShaver from BeOS and the Mac-On_Linux project. The ApplixWare suite of office productivity tools (word processing, spreadsheet, database, graphics) is also available for LinuxPPC.

LinuxPPC runs on PCI-based Power Macintosh computers and compatibles as well as PowerPC BeBox, PReP and CHRP machines. Machines that can run LinuxPPC include:

Manufacturer	Model
Apple Computer	G3 desktop and Powerbooks, iMac, Blue & White G3, 9600, 9500, 8600, 8500, 8200, 7600, 7500, 7300, 7200, 6500, 6400, 6360, 5500, 5400, 4400, Powerbook 2400 and 3400, 20th Anniversary Macintosh
Be	BeBox
IBM	RS6000 (PowerPC-based), 830, 850, 40P, Nobis, INDI
Motorola	StarMax (and all StarMax clones from APS, PowerTools, Mactell), PowerStack, Series E, PowerStack II
Power Computing	PowerBase, PowerWave, PowerCenter, PowerCenter Pro, PowerTower, PowerTower Pro
Umax	C500, C600, J700, S900, Apus 2000 and 3000

Newsgroup

comp.os.linux.powerpc is the newsgroup for Linux on PowerPC, which includes MkLinux, Linux-pmac, and LinuxPPC.

Mailing List

To subscribe to the LinuxPPC-user mailing list, send e-mail to linuxppc-user-request@lists.linuxppc.org with the word **subscribe** in the body.

To unsubscribe, send e-mail to the same address with the word **unsubscribe** in the body.

The traffic on this list can get heavy at times, so you may want to subscribe only to the announcement list for kernel updates and other info. Send an e-mail to linuxppc-announce-request@lists.linuxppc.org with the word **subscribe** in the body.

Archives and search engines for these lists are available at http://lists.linuxppc.org/. There is a developer mailing list available there as well. See http://lists.linuxppc.org/ for more information.

The FAQ-O-Matic

http://www.dartmouth.edu/cgi-bin/cgiwrap/jonh/lppc/faq.pl
An excellent site by Jon Howell, the FAQ-O-Matic contains general info and answers to many common questions concerning LinuxPPC, Linux-pmac, and MkLinux. If individual users have any corrections or additions to contribute, they can easily add them to the FAQ-O-Matic, as it is automatically updated as soon as you submit changes.

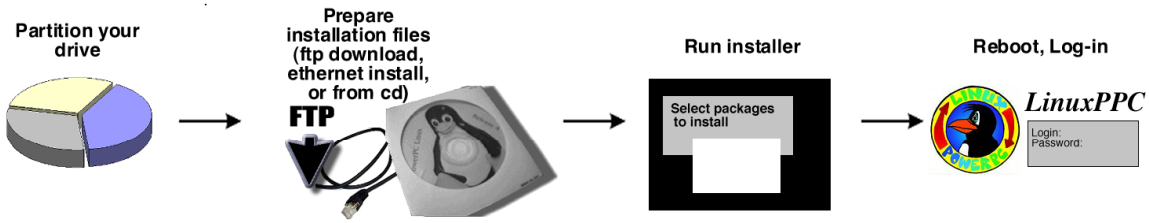
Installation Overview

This section describes:

- Overview of installation
- How much drive space you need

Overview

A brief nutshell of installation:



This guide is focused on installation of Release 5 of LinuxPPC. Resources that document installation not specific to the PowerPC are located at the Linux Documentation Project (http://metalab.unc.edu/LDP/).

LinuxPPC can be installed either using an X-based installer or a Red Hat InstallerThe X-based installer is the recommended method of installing R5, while the Red Hat installer is the traditional method of installation used in previous releases.

The installer runs off of a Linux kernel. The kernel may be booted from a floppy, or for Mac users it can run off of a disk image on the hard drive when used with BootX. It reads the necessary installation files from one of four places:

- CD-ROM (available from https://order.linuxppc.com/)
- FTP via ethernet (with ethernet cards that the installer can recognize)
- NFS via ethernet (with ethernet cards that the installer can recognize)
- an existing local Mac OS HFS disk partition (HFS+ is not yet supported)

On Power Macintoshes and clones, LinuxPPC may be booted either directly from Open Firmware or using BootX. In general, installing and booting using BootX is significantly easier. This guide will only cover installation with BootX, which requires Mac OS to be installed. Documentation on using Open Firmware will be forthcoming.

BootX and Open Firmware: A Comparison	
BootX	**Open Firmware**
BootX is a Mac OS utility from Benjamin Herrenschmidt that can be used in the form of an extension or as an application. The BootX extension provides a simple graphical interface for you to choose between booting Mac OS or LinuxPPC at startup. It also allows the user to choose different boot-up partitions for LinuxPPC or to boot off of a ram disk (such as the one used for the installer). It provides the added capability, to use Mac OS initialized video if no video	Open Firmware is software built into all Power Macintoshes since the 7200. If you are unable to use BootX, then you can use Open Firmware to boot into LinuxPPC. Open Firmware builds a device tree of the computer's hardware and allows control of which operating system to boot. In order to make Open Firmware boot LinuxPPC rather than Mac OS, you will have to modify Open Firmware's boot variables. Boot variables are stored in non-volatile ram, or nvram.

driver exists for your system (currently only ATI and ixMicro Twin Turbo video have built-in drivers). Usually, ATI video is built into Powermacs. The BootX application does the same thing as the extension, except it can be used to quit Mac OS and launch LinuxPPC from the Finder. Using BootX will be covered in more detail in the sections describing the installation and boot-up process.	These settings can be reset to their default values using the following method: at restart, hold down the Command-Option-P-R key combination until the computer resets itself twice (note that this will also clear your PRAM settings, such as settings in the memory control panel, volume, and video settings). Setting boot variables using the Boot Variables application will be covered in an appendix (not yet available).

Amount of Disk Space Required

For default installation, you will need about 500 MB of drive space on its own partition(s) dedicated to LinuxPPC. These partitions may be on the same drive as another OS or on their own dedicated drive. If you want a bare minimum setup (no compilers, no X windows) you can get away with around 70 MB, but its usefulness will be limited. For Release 5, 1.0 GB is recommended.

It is recommended that your drive be split into at least 2 partitions: a root partition and a swap partition. Root is where the Linux system files are kept, swap is used for virtual memory. The size of swap depends on the load you expect on your system. For most single-user systems, 50 MB should be fine.

For Release 5, the recommended partitioning scheme is to split a 1 GB drive into 950 MB for root and 50 MB for swap. You can choose to create separate partitions for some of the larger directories that reside in the root directory, such as /usr, /home, and /var. Keeping these directories in separate partitions is a good idea to help protect your data in the rare case of file system corruption or other problems. If you have to reformat, you will only have to reformat that one partition. If you choose not to create separate partitions for these directories, all the files will be placed in directories in root.

If you will be partitioning a drive that has existing data on it (such as your startup disk), back it up. Editing the partition table of a drive usually results in the loss of everything in the partitions modified. There are some third party disk utilities that will allow you to resize a partition so that existing data is not erased, but use of these utilities will not be covered here.

If you plan on downloading the installation files onto a Mac OS drive, you'll need to partition your disk first. After partitioning, you can then reinstall Mac OS and download the installation files onto your Mac OS HFS partition (HFS+ is not yet supported). Alternatively, if you have two drives, you can retain the first drive for Mac OS and partition the second for LinuxPPC installation files. If you are installing from the CD, NFS, or FTP, you do not have to worry about copying the files onto a HFS partition.

More details on how to split your drive will be covered in the next chapter.

Partitioning the Hard Drive

This section describes:

- Backing up
- Partitioning schemes
- Partitioning programs

Before you do any partitioning, remember to backup all important files on a separate hard drive, since partitioning a disk typically makes all old data on that disk inaccessible. If you have a brand new Apple machine that includes the "Apple Software Restore" application on the system CD-ROM, a backup won't be necessary if you haven't installed any additional files on your machine.

Partitions come in several types: Apple Hierarchical File System (Apple_HFS, used by Mac OS), HFS+ (also type Apple_HFS, used by Mac OS 8.1 and above), Be File System (Be_BFS), and A/UX (Apple_UNIX_SVR2), among others. LinuxPPC requires A/UX partitions.

For a beginner coming from Mac OS, it may be easiest to do the initial partitioning with Apple's Drive Setup program so that you get the sizes right and have the proper drivers for partitions used with Mac OS. Although Drive Setup cannot create A/UX partitions, you can create HFS partitions that can be easily converted to A/UX partitions with the LinuxPPC installer using pdisk (also referred to as fdisk). Users of other operating systems may go straight into using pdisk to partition the drive.

Pdisk is a partitioning program that runs within Linux. If you choose to partition directly with pdisk, you can skip the remainder of this chapter and go to the next chapter on Installation Setup. There is also a Mac OS port of pdisk that runs as a command-line-interface application available at ftp://ftp.linuxppc.org/pub/linuxppc/linuxppc-R4/RedHat/tools/. If you have problems using the installer's version of pdisk, you can try using the Mac OS version.

The remainder of this chapter goes over creating HFS partitions with Drive Setup and converting them to A/UX using the installer. If you feel comfortable with partitioning, you can skip using Drive Setup and use pdisk to directly create A/UX partitions, but remember that pdisk does not write Mac OS-specific drivers for HFS partitions it creates. Pdisk is quite flexible once you are familiar with using it, and instructions on using pdisk are located in Appendix D.

Commercial disk utilities such as Hard Disk Toolkit, CharisMac Anubis, and Silverlining can also create A/UX partitions. However, use of these utilities will not be covered here.

If you are trying to partition a SCSI hard drive, you can also use Apple_HD_SC_Setup_7.3.5 in Mac OS (available from ftp.info.apple.com for free) to create A/UX partitions. If the drive does not include an Apple ROM, you might have to modify HD_SC_Setup with ResEdit so that it will recognize your drive (change the wfwr resource from 00 to FF).

As I mentioned before, major directories within root can be separate partitions or part of the main root partition. If you are planning to use LinuxPPC as a multi-user server, you may also consider creating a separate partition for /home (where all user accounts are stored).

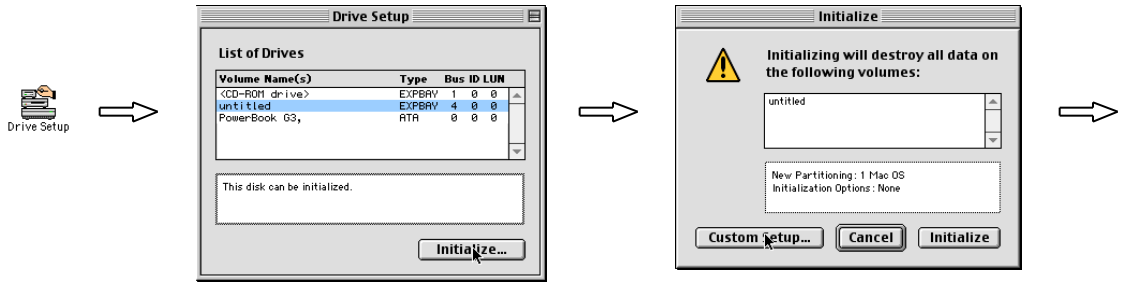
Swap must be a separate partition from root. The size of swap should be around 50 MB, or more if you expect your system to have a heavy load as may be the case if your system is going to be used for a server.

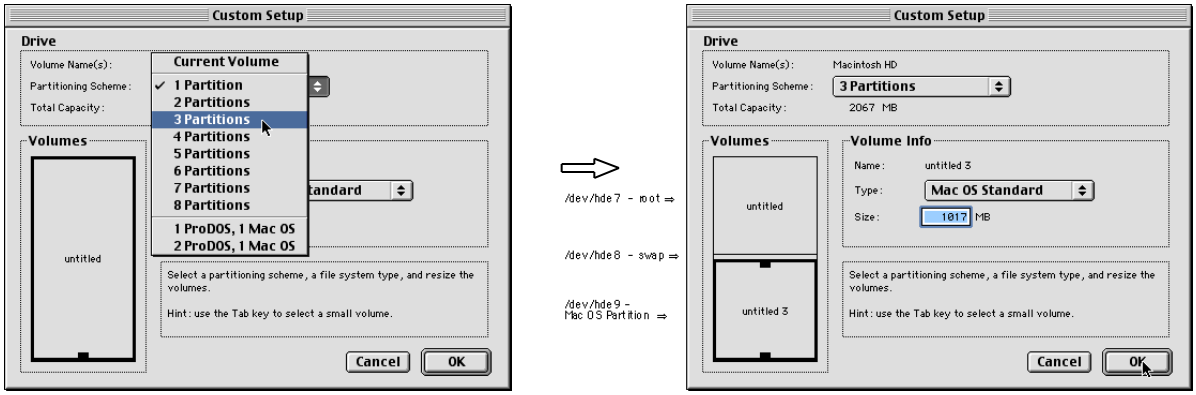
LinuxPPC Release 5 recommends at least 500 MB for a default installation, and 1 GB if you choose to install everything.

During partitioning, remember the order your partitions are in - Drive Setup uses the first few partitions of the drive for various drivers, patches, and partition maps, so your usable partitions might start numbering from partition 4, 5 or higher. Linux uses the following scheme for accessing partitions on an IDE drive: /dev/hdXY, where X is the device ID (a letter like "a," "b," "c," etc.) that the kernel assigns to the drive and Y is the number of the partition on that drive. For example, the first usable partition on an IDE drive might be /dev/hda7. For SCSI drives, the only difference is that partitions are accessed using the /dev/sdXY scheme. The internal SCSI drive is usually /dev/sdaY.

In the example here we will partition an IDE drive. Remember that you cannot partition the disk that you booted from - you will have to launch Drive Setup from the Mac OS CD, a Disk Tools floppy, or a separate drive. You can also download Drive Setup from Apple. Be aware that Drive Setup will format the partitions once they've been created, erasing all data, so backup your stuff. As mentioned earlier, if you have a brand new machine from Apple and haven't installed any new files, you can utilize the "Apple Software Restore" program found on the system CD-ROM to reinstall all the original files.

Drive Setup can normally be found in the Utilities folder. After launching Drive Setup, select the drive you want to partition, and click on the "Initialize…" button. Then click on the "Custom Setup…" button. This will bring up the custom partitioning window.

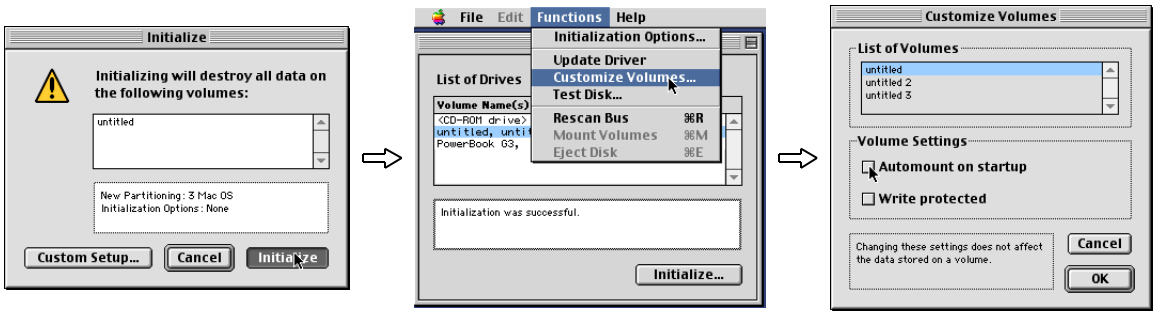




Here we selected the 3 partition scheme, and customized the size of each partition. You can utilize the tab key to select each partition and type in its size. The first shown partition is root, which is actually the seventh partition overall – Drive Setup keeps the partitions for the partition maps and drivers hidden. Thus, root is at /dev/hda7, swap is at /dev/hda8, and the partition used for Mac OS will be at /dev/hda9.

note: at this time, LinuxPPC does not recognize HFS+ drives. If you're creating partitions that will be converted to A/UX later, use HFS (Mac OS Standard). If you will be downloading the installation files to a hard drive partition, that partition must use standard HFS for the installer to find them. Don't forget to reserve space on your Mac partitions to download the installation files if you're installing from the hard drive (this is not necessary if you're installing from the LinuxPPC CD, NFS, or FTP). You will need around 500 MB of drive space for the installation files (and over 850 MB if you are installing everything).

When you're done, click on the "Okay" button, then click the "Initialize" button. After partitioning is complete, select the drive again from the list and go to the "Functions" menu and select "Customize Volumes…". This will bring up a window that allows you to select each individual partition. Select your Linux partitions and uncheck the "Automount on startup" option.



When you have disabled automount on each of your Linux partitions, click OK and quit Drive Setup. If you have problems using Drive Setup, please refer to Apple's documentation in the Help menu.

If you're planning on sharing the same drive with Mac OS, now is a good time to reinstall Mac OS on one of the partitions you're not using for LinuxPPC. You can also utilize the "Apple Software Restore" application that came on the CD's of recent Power Macintoshes to restore Mac OS to its original state. In this example, I reinstalled Mac OS from backup onto the partition I created for Mac OS (partition #10). The Mac OS partition will also be used for the LinuxPPC installation files (again, this won't apply for those doing CD, NFS, or FTP installs).

Continue to the Installation Setup chapter to see what files you will need and where to get them.

Installation Setup

This section describes:

- Where to get the installation files
- What files you need
- Customizing the installation (not yet available)

The files you need for installation can be broken down into two categories:

- Installation files required for all operating systems
- Specific files for your system.

The easiest way to install is to get the LinuxPPC CD-ROM, which includes all the files you need. For Mac users, the CD-ROM takes advantage of BootX and LinuxPPC Live to bring you to an X Windows-based installer at the click of an icon. If you have the LinuxPPC CD-ROM, you can skip this chapter and go on to the next chapter on installation. If you choose to download the files, they must be downloaded in binary (raw) format. Remember that it must be a HFS drive, since HFS+ drives are not yet supported. The complete download is nearly 500 MB, or at least 200 hours of download time for you dial-up modem users `:-)` - a LinuxPPC R5 Lite should be forthcoming. I'll be updating this guide in the future to show you how to customize an installation so that you can reduced the number of files you need to download.

LinuxPPC 1999 (R5) CD-ROM

The LinuxPPC CD-ROM can be ordered securely at https://order.linuxppc.com/. Purchasing the CD-ROM helps support the LinuxPPC Project. LinuxPPC, Inc. uses portions of the proceeds from CD orders to support LinuxPPC developers and to provide systems to commercial developers so that they can provide ports to LinuxPPC. Funds are also used to support the LinuxPPC.org web site and ftp sites.

Download Sites

The main ftp site for LinuxPPC R5 is at ftp://ftp.linuxppc.org/linuxppc-R5-final. The following is a list of some mirror sites for LinuxPPC – please use a site that is geographically close to you. The most up-to-date list of mirrors is located at http://www.linuxppc.org/mirrors/.

```
North American Mirrors:

 CA - ftp://ftp.cdrom.com/pub/linux/linuxppc/
 IL - ftp://uiarchive.uiuc.edu/pub/systems/linux/linuxppc/
 MA - ftp://rufus.w3.org/linux/linuxPPC/
 MI - ftp://ftp.eecs.umich.edu/pub/linux/linuxppc
 MO - ftp://wuarchive.wustl.edu/systems/linux/linuxppc/
 NC - ftp://sunsite.unc.edu/pub/Linux/distributions/linux-ppc/
 TN - ftp://sunsite.utk.edu/pub/linux/LinuxPPC/
 WI - ftp://mirror.doit.wisc.edu/pub/mirrors/linux/distributions/linuxppc/
 WI - ftp://dev.linuxppc.org/pub/linuxppc/
 CAN - ftp://ftp.linuxberg.com/pub/distributions/

European Mirrors:

 AT - ftp://gd.tuwien.ac.at/opsys/linux/linuxppc/
 CH - ftp://sunsite.cnlab-switch.ch/mirror/linuxppc/
 DE - ftp://ftp.apfel.de/pub/LinuxPPC/
 DE - ftp://ftp.uni-bremen.de/pub/linux/dist/linuxppc/
 ES - ftp://ftp.fi.upm.es/pub/mirrors/linux/linuxppc/
 FI - ftp://ftp.funet.fi/pub/mirrors/ftp.linuxppc.org/
```

```
 FR - ftp://ftp.lip6.fr/pub/linux/linuxppc/
 NO - ftp://mac.pvv.ntnu.no/pub/linux/linuxppc/
 SE - ftp://ftp.karen.hik.se/pub/linux/linuxppc/
 UK - ftp://sunsite.doc.ic.ac.uk/Mirrors/ftp.linuxppc.org/pub/linuxppc/

Australian Mirrors:
 - ftp://ftp.au.linuxppc.org/pub/linux/linuxppc/
 - ftp://mirror.dstc.edu.au/pub/linuxppc/
 - ftp://the.ausmac.net/pub/mac/Linux/LinuxPPC/
 - http://the.ausmac.net/ftp/Linux/LinuxPPC/

Japanese Mirrors:
 - ftp://mirror.nucba.ac.jp/mirror/linuxppc/
 - http://mirror.nucba.ac.jp/mirror/linuxppc/
 - ftp://ftp.ring.gr.jp/pub/linux/linuxpp/
 - http://www.ring.gr.jp/pub/linux/linuxpp/
 - ftp://sunsite.sut.ac.jp/pub/archives/linux/linuxppc/
 - http://sunsite.sut.ac.jp/pub/archives/linux/linuxppc/
 - ftp://ftp.ccex.miyazaki-u.ac.jp/pub/ftp.linuxppc.org/
 - ftp://ppc.linux.or.jp/pub/mirrors/LinuxPPC/
```

A Note About FTP

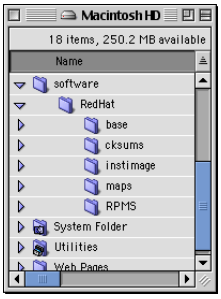
You can use your favorite FTP program to download the files, but for Mac OS users I recommend using Fetch to download binaries. Netscape and Internet Explorer should be able to download the necessary files as well, but be sure that they aren't downloaded in text format. Netscape and IE users should configure their browsers so that file types with extensions rpm, coff, tgz, and gz are downloaded in binary. In Netscape, this is in the Preferences > Navigator > Applications setting. In IE, this is in the Preferences > Receiving Files > File Helpers setting. You can get Fetch from the Info-Mac hyperarchive: http://hyperarchive.lcs.mit.edu/HyperArchive/Archive/comm/inet/fetch-303.hqx.

Files Necessary for All Systems

You only need to follow the steps in this section if you will be downloading all the installation files onto your hard drive for installation. If you are performing the installation directly through FTP or NFS in the installer, you can go on the next section on "Specific Files for your System." Again, those performing an installation via the LinuxPPC CD-ROM can skip on to the next chapter on using the installer.

Remember that to download the installation files you'll need around 500 MB of free drive space. The bulk of the LinuxPPC installation is in RPM format, approximately 420 MB if you download everything. RPM stands for Red Hat Package Manager, and is the format used for installation of most LinuxPPC programs. RPM packages for LinuxPPC are designated by the .ppc.rpm extension (these can also be used by MkLinux).

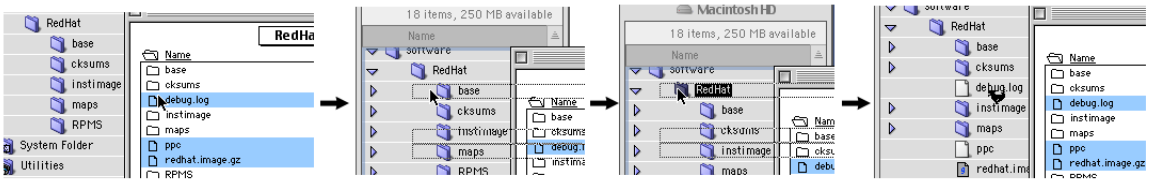
To set up, you will need a folder/directory called "software" in the root level of your hard drive. For Mac users, double-click on the icon of your hard drive. Create a new folder in the hard drive window, and rename it software.



Next, create a new folder/directory called "RedHat" in the "software" folder, and then create folders called "base", "cksums", "instimage", "maps", and "RPMS" inside the "RedHat" folder. If you use Mac OS, you should end up with something like the image to the right.

Next, go to one of the mirrors above and into the linuxppc-R5-final/RedHat directory. I will use ftp.cdrom.com and the /pub/linux/linuxppc/linuxppc-R5-final/RedHat/ directory as an example.

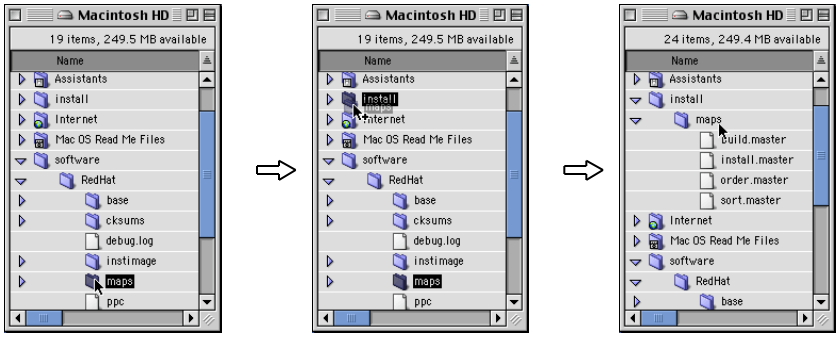
First you will need to copy the debug.log, ppc, and redhat.image.gz files into the "RedHat" folder on your hard drive. If you're using Fetch, the easiest thing to do would be to command-click the individual files to select them and drag them all onto the RedHat folder on your drive.



For the remaining files in each of the base, cksums, instimage, maps, and RPMS directories, it's safest to go into each directory individually, select all, and copy them all into the individual folders of your hard drive. We do this rather than copy the directory itself because the files in each directory are often symbolic links to files located elsewhere on the mirror and may not get resolved properly by your FTP client. For example, go into the base directory, select all files, and copy them into the base folder on your hard drive (in Fetch, you can use command-A to select all inside the base directory and drag-and-drop onto the base folder). Repeat this process for the remaining directories in the RedHat folder. Remember the RPMS folder contains about 420 MB worth of rpm's. I will be updating this guide in the future to show you how to customize the installation so you only download the rpms you need.

Workaround for limitations of the Mac OS 31-character filename limit:
Some rpm files that you download may contain more than 31 characters in their filenames. As a result, your FTP client will shorten these filenames so that they fit under Mac OS's 31-character filename limitation. Different FTP clients will shorten the filenames in different ways – Fetch simply cuts off the last few characters of the filename, whereas Anarchie retains the last four characters (.rpm), cuts off some of the middle characters and replaces them with an ellipses (…). As a result, the X-based installer may not properly find these files. The workaround is to download a modified hdlistinfo file and replace the one located inside the software > RedHat > base folder. You can download a modified hdlistinfo for Fetch at http://www.linuxppc.org/userguide/hdlistinfo.fetch.sit, which you should unstuff, rename to hdlistinfo, and place it inside the base folder. A similar hdlistinfo is available at http://www.linuxppc.org.userguide/hdlistinfo.anarchie.sit for users that downloaded the RPMS with Anarchie. Special thanks to LinuxPPC users Drew Thoeni and Fabian Ille for providing these files. Note that the Red Hat installer does not have this problem with Mac OS-shortened filenames.

Finally, you need to create a folder/directory called "install" in the root level of your hard drive. Then copy the "maps" folder into it. For Mac users, the easiest way to do this is to hold down the option key, click-and-hold onto "maps," and drag it onto the "install" icon. You should end up with the files build.master, install.master, order.master, and sort.master inside.

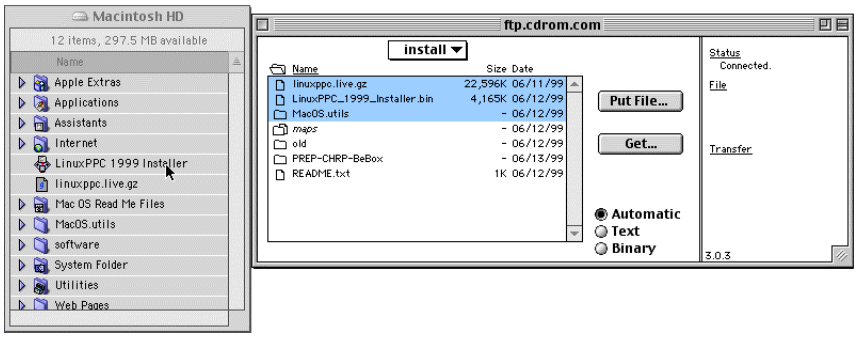


Specific Files for Your System

Specific files for each system can be found the LinuxPPC-R5-final/install directory in any of the mirrors above. See the section below that pertains to your system.

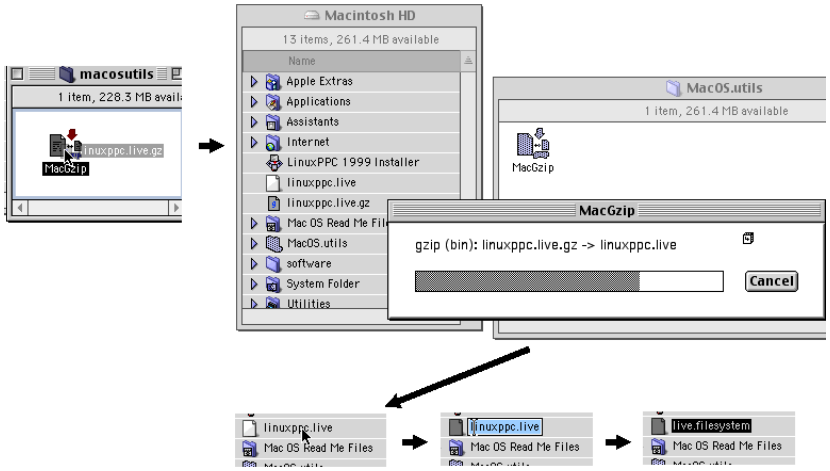
Mac OS systems

Once you get to the install directory, select the linuxppc.live.gz, LinuxPPC_1999_installer.bin, and the MacOS.utils directory and drag it into your hard drive (on some FTP servers, linuxppc.live.gz is named live.filesystem.gz).



The linuxppc.live.gz (live.filesystem.gz) file contains the LinuxPPC Live file system, which provides the X Windows-based installer. It is the recommended method of installation for R5, although you can use a Red Hat-style installer instead if you'd like to save yourself the download. The boot-up scripts will revert to the Red Hat-style installer if a live file system cannot be found. The CD-ROM will use the X-based installer be default. You can also enter "redhat" in the kernel arguments when you use BootX or Open Firmware to launch the installer (more about this later).

Next, open the MacOS.utils folder and drag-and-drop the linuxppc.live.gz (live.filesystem.gz) file onto MacGzip – this will ungzip the live file system. If necessary, rename the linuxppc.live file to live.filesystem after ungzipping is complete.



Next, double click the LinuxPPC 1999 Installer icon.

This will install ramdisk.image.gz (a ram disk image that the installer first uses) into the System Folder, a Linux Kernels folder into the System Folder, the BootX extension in the Extensions folder, BootX Settings in the Preferences folder, and the BootX application (labeled as "Boot LinuxPPC") onto your desktop. Inside the Linux Kernels folder is a file called LinuxPPC Standard – this is the LinuxPPC kernel, also

called vmlinux. It's the core of the operating system, analogous to the System file in Mac OS. More pre-compiled kernels can be found in the linuxppc-R5-final/kernels directory in the LinuxPPC mirrors. If you have problems booting, you might want to try one of the kernels located there (there are special kernels for the Blue & White G3's, Apple Network Servers, and machines with Adaptic 2940 UltraWide SCSI cards there). You can download these kernels and place them inside the Linux Kernels folder (they must be ungzipped first).

After running the Installer, you will be given the option to reboot. Go ahead and do so – after the Mac OS splash screen, you'll get the BootX screen, which will already be configured properly. Go on to the next chapter.

BeBox, CHRP, and PReP System Setup

Currently the main part of this guide has little specific information for installing LinuxPPC on BeBox, CHRP, or PReP systems. I hope to get a PReP machine in the so that I can improve this aspect of the guide. You can find the system-specific files for these systems in the linuxppc-R5-final/install/PREP-CHRP-BeBox/ directory.

BeBox – one of the best sources of information for LinuxPPC on the BeBox is at http://www.guru.dircon.co.uk/belinux/status.html, a website by William R Sowerbutts
CHRP – CHRP system users can refer to http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/, a web site by Geert Uytterhoeven.
PReP - Kazunori Aoshima has created a web site for installation onto PReP machines at http://ppc.linux.or.jp/~aoshimak/install.html

Now that you have all the files you need, you can start with installation.

Using the Installer

This section describes:

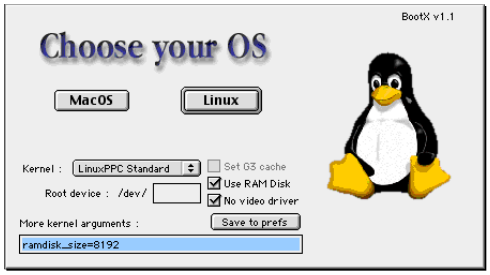
- Starting BootX to boot the installer
- Using the X Linux Installer (XLI), version 1.x (version 2.x will be available later)
- Using the Red Hat Installer (only if you choose not to use the X installer)

Notes for Installing from CD-ROM for Mac Users

If you are installing from CD-ROM, insert the LinuxPPC CD into the drive. You should run the LinuxPPC 1999 Installer, which will place the required files onto your hard drive (refer to the previous chapter to see what files are installed). If you have an older print CD-ROM without the LinuxPPC 1999 Installer, you can download it from the LinuxPPC-R5-final/install directory of any of the mirrors listed in the previous chapter. Otherwise, you can accomplish the same thing as the installer by doing the following: place the BootX settings from the CD into your Preferences folder, copy the Linux Kernels folder and ramdisk.image.gz file from the CD into your System Folder, unstuff the BootX_1.1.sit file located in the macosutils folder of the CD, move the BootX extension from the unstuffed BootX folder into your Extensions folder, and copy the Install LinuxPPC icon onto your desktop.

BootX (Mac OS only)

Users that ran the LinuxPPC 1999 Installer will be greeted with the BootX screen upon reboot.



It will start an automatic boot countdown – you can disable the countdown by hitting tab. Click on the Linux button to continue loading the installer.

Notes on using BootX:

- You can use the tab key to toggle between Mac OS and Linux.
- The Kernel: menu allows you to select which kernel to boot from the Linux Kernels folder inside the System Folder
- The “Root device” space is left blank for now. After you have installed LinuxPPC, you would input the location of your root partition in this space.
- The “Set G3 cache” option allows you to set-up the state of the cache on G3 systems using G3 Cache Profiler by PowerLogix. Note that the G3 cache should already be configured properly for Apple G3 systems, but the cache control can be useful for systems with G3 upgrade cards. G3 Cache Profiler can be downloaded from http://www.powerlogix.com/support/software.html.
- The “Use RAM Disk” option is only necessary the first time you boot the installer – it loads the ramdisk.image.gz located on your hard drive or CD.
- The “No video driver” option uses Mac OS initialized video (unaccelerated). For now it is recommended that you keep this enabled – you can try disabling it and passing video settings to the kernel once you are more comfortable with LinuxPPC (covered in the User Guide).
- Clicking on the “Save to prefs” button will store these settings so that you can select your default OS and store your other settings

- The “More kernel arguments” field includes specific information that you would like to pass to the kernel – for now you should leave it with the default setting (this setting increases the amount of ram that the kernel sets aside for the ram disk). Again, this is only necessary for initial installation and can be left blank once you have installed. This is also the location where you can specify video settings, the run level you boot into, and other options. More about this later. If you want to use the Red Hat style installer, you should append “ redhat” (without the quotes, with the space before redhat) to the end of the argument.

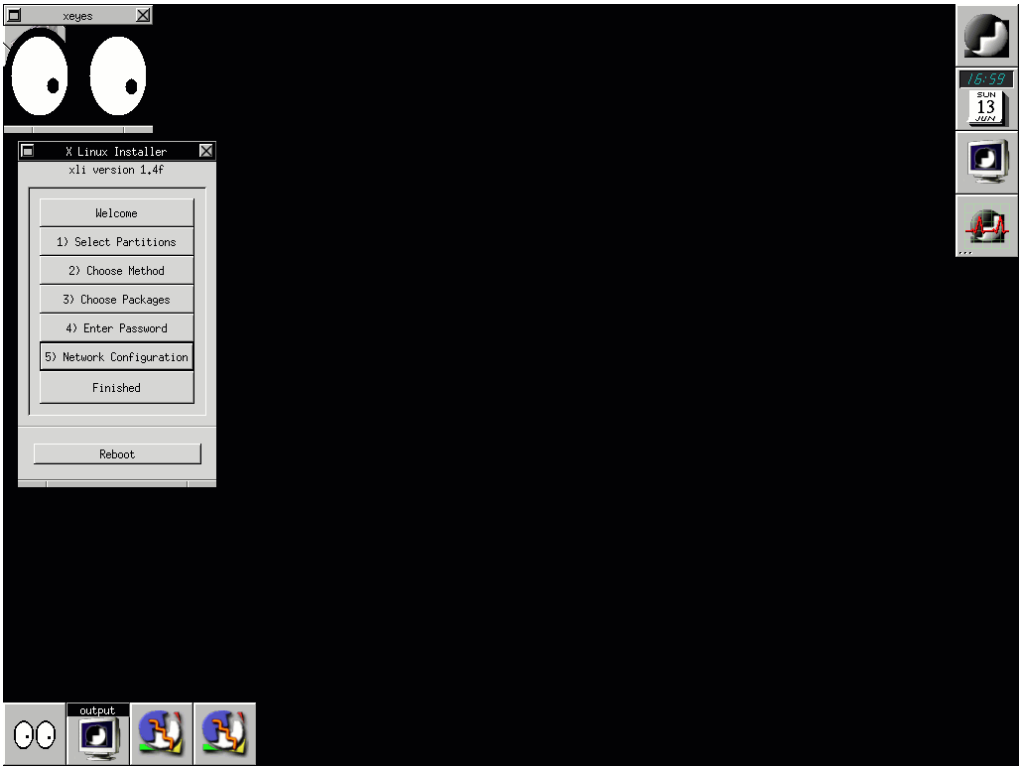
Loading the Kernel

As the kernel loads, the screen will turn black and you will see the Linux penguin in the corner. Text will scroll by on the screen as the kernel proceeds to load.

The X Windows-based Installer

The installer will then check the CD-ROM for the live.filesystem file so that it can load the X Linux Installer. If you are not installing from CD-ROM, the installer will look through your hard drive partitions (except for HFS+ partitions) for live.filesystem. If it’s loading from CD, it will take a few minutes. Loading from the hard drive will take just a few seconds.

After the installer loads, you will be greeted by the following screen (choose method button will not be available if the installer finds the CD-ROM or finds the correct directory structure on your hard drive):



The X Linux Installer’s main interface is on the left side of the screen. Start with the “Welcome” button. Although it states that it cannot format HFS partitions, you can go into pdisk through an XTerm window and change the partitions types from HFS to AUX. One other note: in the lower left-hand corner you’ll notice an icon with the word “output” above it – this is a minimized XTerm window that displays the output from the installer. If you’d like to monitor installation progress, you can open up this window by double-clicking on the icon. It is particularly useful if you are having problems and would like to troubleshoot.

Bringing Up pdisk in an Xterm Window
Click the mouse somewhere on the background of the screen – this will bring up the menu on the right.

Select XTerm – this will bring up a command-line Xterm window where you can type in "pdisk /dev/hda" followed by return to bring up pdisk (substitute /dev/hda with the actual name of your hard drive, typically /dev/hda for an internal IDE drive or /dev/sda for the internal SCSI). Second internal IDE hard drives would be /dev/hdb if it's a slave drive or /dev/hdc if it's on the second bus. Secondary SCSI drives would be /dev/sdb, /dev/sdc, etc. based on the order that the kernel finds them on the SCSI bus. On a PowerBook G3, the right expansion module is /dev/hdc, and the left module is /dev/hde.

Here are the relevant commands in pdisk that you'll need:

?	Help
p	print out your partition map to screen
d [#]	deletes partition number [#], converts it to free space. Example: d 5 will delete partition #5
c	creates a new Apple_UNIX_SVR2 (A/UX) partition
w	writes the partition map to the disk
q	quit - does not automatically save changes, must use w first

This is what you see when you first call up pdisk (my typed-in input in red – note that I am editing a expansion module drive on a PowerBook, so I'm using /dev/hde):

```
[root@(none) ~]# pdisk /dev/hde
Edit /dev/hde –
Command (? for help):
```

I then type in "p" to print out the partition map. Notice that the first few partitions are the partition map and drivers. It also lists the length, in terms of 512 byte block sizes, of each partition and the base block that it starts in.

```
Command (? for help): p

Partition map (with 512 byte blocks) on '/dev/hde'
 #:                type name             length   base    ( size )
 1: Apple_partition_map Apple                63 @ 1
 2:      Apple_Driver43*Macintosh            54 @ 64
 3:      Apple_Driver43*Macintosh            74 @ 118
 4:    Apple_Driver_ATA*Macintosh            54 @ 192
```

```
 5:    Apple_Driver_ATA*Macintosh            74 @ 246
 6:       Apple_Patches Patch Partition     512 @ 320
 7:           Apple_HFS untitled        2048000 @ 832     (1000.0M)
 8:           Apple_HFS untitled 2       102400 @ 2048832 ( 50.0M)
 9:           Apple_HFS untitled 3      2082358 @ 2151232 (1016.8M)
10:          Apple_Free Extra               10 @ 4233590

Device block size=512, Number of Blocks=4233599 (2.0G)
DeviceType=0x0, DeviceId=0x0
Drivers-
1: @ 64 for 22, type=0x1
2: @ 118 for 36, type=0xffff
3: @ 192 for 20, type=0x701
4: @ 246 for 33, type=0xf8ff
```

Notice that partitions 7, 8, and 9 are of type “Apple_HFS” – these are the partitions I created earlier in Drive Setup. Partition 7 will be used as root and partition 8 will be used as swap – these will both have to be converted to AUX to be used with Linux. I’ll keep partition 9 as HFS so that I can transfer files between Linux and Mac OS. I’ll start by deleting partition 7 so that it is no longer HFS, then recreating it as AUX.

```
Command (? for help): d 7
Command (? for help): c 7p
Length in blocks: 2048000
Name of partition: root
```

The “d” command followed by the partition number will delete that partition, so I used “d7” in this case. The “c” command followed by the partition number appended by the letter “p” creates a new partition starting at that partition (“c 7p” in this case). I then gave it the size of the partition length in blocks, which I knew was 2048000 from printing out the partition earlier. Finally, I gave it the name root.

I then repeat the process with the partition #8 for swap. If there are other partitions that you would like to use as /home, /usr, etc. then repeat the process for those partitions as well.

```
Command (? for help): d 8
Command (? for help): c 8p
Length in blocks: 102400
Name of partition: swap
```

Finally, to check that everything is okay, I print out the partition map again:

```
Command (? for help): p

Partition map (with 512 byte blocks) on '/dev/hde'
 #:                type name             length   base    ( size )
 1: Apple_partition_map Apple               63 @ 1
 2:      Apple_Driver43*Macintosh           54 @ 64
 3:      Apple_Driver43*Macintosh           74 @ 118
 4:    Apple_Driver_ATA*Macintosh           54 @ 192
 5:    Apple_Driver_ATA*Macintosh           74 @ 246
 6:       Apple_Patches Patch Partition     512 @ 320
 7:     Apple_UNIX_SVR2 root            2048000 @ 832     (1000.0M)
 8:     Apple_UNIX_SVR2 swap             102400 @ 2048832 ( 50.0M)
 9:           Apple_HFS untitled 3      2082358 @ 2151232 (1016.8M)
10:          Apple_Free Extra               10 @ 4233590


Device block size=512, Number of Blocks=4233599 (2.0G)
DeviceType=0x0, DeviceId=0x0
Drivers-
```

```
1: @ 64 for 22, type=0x1
2: @ 118 for 36, type=0xffff
3: @ 192 for 20, type=0x701
4: @ 246 for 33, type=0xf8ff
```

It all looks good, so I use the "w" command to write the partition map. If you messed up, you can simply quit pdisk using "q" without writing the partition map and start it back up using pdisk /dev/hda (substituting the correct name of the drive).

```
Command (? for help): w
Writing the map destroys what was there before. Is that okay? [n/y]: y
```

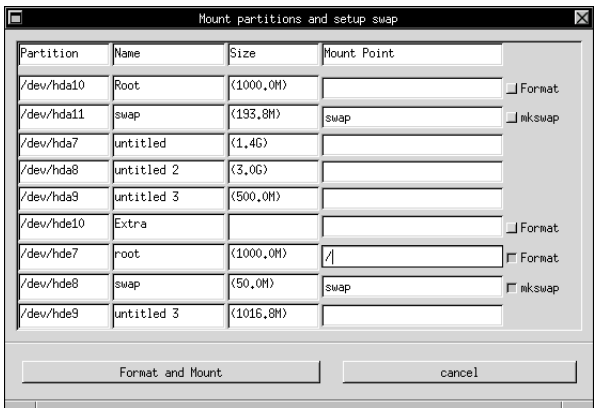
You may get the following error message:

```
Re-read of partition table failed  (Device or resource busy)  Reboot your system to ensure
the partition table is updated
```

Type "q" to quit, then type "reboot" in the Xterm window – this will reboot into Mac OS, and bring you to the BootX login screen. Make sure that "Use Ram Disk" is still checked and that "ramdisk_size=8192" is still in the "More kernel arguments" field, then click on the "Linux" button. This will bring you back into the installer.

Select Partitions

Click on the "Select Partitions" button to bring up the "Mount partitions and setup swap" window. In this case I only want to install onto /dev/hde7 as my root partition and /dev/hde8 as my swap space (the partitions at /dev/hda10 and /dev/hda11 are from a previous R4 installation and won't be used here).

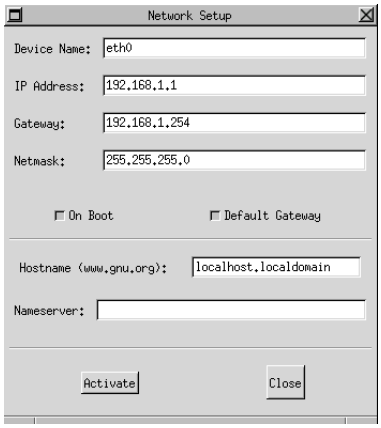


So under the Mount Point column I entered "/" for /dev/hde7 and indicated that I wanted that partition formatted by clicking on the button next to it. Write down the ID of this partition – you will need it later in BootX to specify the location of root. Swap was already entered since the installer recognized "swap" as the name of the partition. If you have any other partitions that you'd like to use as /home, /usr, etc. then select those partitions and type in the name of the directory you'd like that partition to mount at (for example "/home"). When finished, click on the "Format and Mount" button. It will take a few seconds to perform the operation.

Choose Method

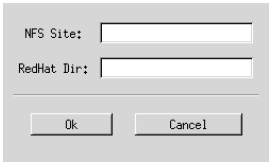
Next, you can select the "Choose Method" button. *Note:* if you started the installer with the LinuxPPC CD in the drive, this button will not appear since the installer has already defaulted to a CD-ROM installation. The same is true if you downloaded all the installation files to the hard drive as described in the previous chapter. Otherwise, you will get the following window after clicking the "Choose Method" button:

If you choose CD-ROM, make sure that you have the CD inside the drive. The NFS, FTP, HTTP selections will require that you set up your network settings:



You can get these settings from your original operating system, talking to your system admin, or by playing with the ifconfig, netstat, and route utilities in an Xterm window (the kernel will do its best to find out on its own your network settings). Mac users can check the Apple System Profiler or the TCP/IP control panel. If your network uses dhcp, you can have your machine query for a dhcp address by going into an Xterm window and typing in dhcpcd. This will give you your IP address and put your name server in /etc/resolv.conf (type in "cat /etc/resolv.conf" to view them).

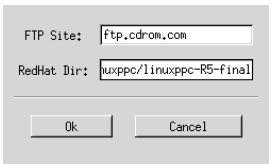
After configuring networks settings for a NFS, FTP, or HTTP installation, you'll get a window that looks something like this (I'm using NFS as an example):



Type in the address of the server followed by the directory that contains the RedHat. For example, if I were to perform an FTP installation from the ftp.cdrom.com mirror, I would input the following:

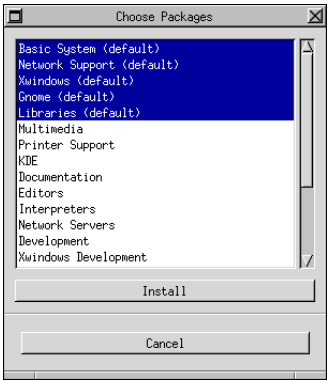
```
ftp.cdrom.com
pub/linux/linuxppc/linuxppc-R5-final
```

Notice that I did not put a / before "pub/linux/linuxppc/linuxppc-R5-final" nor is there a / at the end. It would look like this when I was finished:



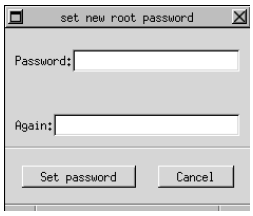
Then click "Ok" – the installer should query the site and download the necessary files for you to proceed to the next step, choosing the packages.

Choose Packages
The installer should then bring up the Choose Packages window automatically – otherwise click on the button from the main menu window.



The default installation will require 350 MB of drive space – if you choose everything, it will consume nearly 1 GB. Then click on Install after selecting the files you want. (I will be updating this guide later to include details on different packages). You'll get a progress bar that indicates how installation is going. If you'd like more details, you can double click the "output" icon to get a Xterm window that displays each RPM being installed.

Enter Password
The next step is to enter the password for root (the superuser of the machine). Don't use any dictionary words, since this is the most critical aspect of security on your system.

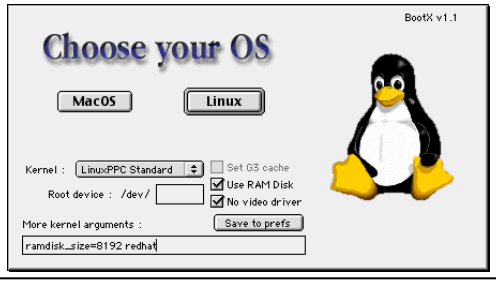


Network Configuration
As described earlier, you can optionally set-up your network settings here.

Congratulations! Installation of LinuxPPC R5 is complete. Click on the "Finished" button to read some notes, then hit Reboot. If you use Mac OS, it will reboot into Mac OS and into the BootX screen. Hit tab to disable the countdown so that you can set up BootX to boot R5. Skip past the next section on using the Red Hat Installer and continue to the chapter on Booting LinuxPPC.

Using the Red Hat Installer

To use the Red Hat Installer instead of the X-based installer, you should append the word " redhat" to the end of the "More kernel arguments" field in BootX. Note that there is a space between the statements "ramdisk_size=8192" and "redhat". Keep in mind that the X-based installer is the recommended method of installing LinuxPPC.



As the installer loads, you may see a message about partitioning and formatting your disk, mounting them on /mnt, and running install.pl – this can be safely ignored as it only pertains to the X-based installer.

Once the installer has loaded, you will get a splash screen for the LinuxPPC Reference Release. The installer is entirely keyboard-driven – the mouse will not be used. You can use the tab and arrow keys to move between different fields, and the space bar to select. The first few screens will ask you for the language that you would like to run the installer in and the type of keyboard that you have.

Installation Method screen

There are five options for the installation method: local CD-ROM, NFS image, hard drive, FTP, and HTTP.

Here are some notes for each path:

- CD-ROM - If you have more that one CD-ROM drive, the installer will try to mount the first drive you have. If the installer can not mount the CD, try one of the other CD-ROM drives you have. Some ZIP drives are known to cause problems with the SCSI chain on some kernels. Try turning off your ZIP drive during the install if you're having problems.
- FTP, HTTP, or NFS- You must use a kernel that recognizes your ethernet adapter. The default LinuxPPC R5 kernel recognizes the built-in ethernet adapter of most PowerMacs. To configure your network, you will need to know the IP address, netmask, gateway, nameserver, domain name, and host name that your system uses. When it asks to select the host, use the address of ftp.linuxppc.org or the mirror nearest you. If the nameserver is not configured properly, use the IP address instead – for example, ftp.linuxppc.org has an IP address of 169.207.161.2. You can use any number of web-based nslookup sites to find the IP addresses of the mirrors listed on pages 11-12. One free nslookup web site is at http://www.citilink.com/cgi-bin/cgiwrap/kae/onestop.pl.
  For the path to RedHat directory, use the absolute pathname of the directory containing - at ftp.linuxppc.org this is “linuxppc-1999” (without the quotes). If the installer did not find your ethernet card, you may have to install from CD-ROM or a local hard drive.
- Local hard drive - the RedHat folder and files must be on an HFS (not HFS extended) partition. If you used the directory structure described in the previous chapter, the directory would be “/software”.

Installation Path

The next screen asks you whether you are installing a new system or upgrading a previous system. Choose the appropriate selection.

Drive Setup

The screen will give you the choice of using Disk Druid or Fdisk (which is the same as pdisk). Disk Druid cannot work with HFS partitions, so if you have HFS partitions you will have to use pdisk. For this reason, use of Disk Druid will not be covered in this guide, though you can find information on using Disk Druid at http://www.redhat.com.

Partition Disks with Fdisk

Select "edit" to go into fdisk to partition your drive or to change your HFS partitions to A/UX. You may want to refer to Appendix D: Partitioning with pdisk and fdisk.

Here are the relevant commands in fdisk that you'll need:

?	Help
p	print out your partition map to screen
d [#]	deletes partition number [#], converts it to free space. Example: d 5 will delete partition #5
n	creates a new Apple_UNIX_SVR2 (A/UX) partition
w	writes the partition map to the disk
q	quit - does not automatically save changes, must use w first

The first step is to print out your partition map to screen with p. You should get something like this:

```
Partition map (with 512 byte blocks) on '/dev/hde'
#:                type name             length   base    ( size )
1: Apple_partition_map Apple                63 @ 1
2:      Apple_Driver43*Macintosh            54 @ 64
3:      Apple_Driver43*Macintosh            74 @ 118
4:    Apple_Driver_ATA*Macintosh            54 @ 192
5:    Apple_Driver_ATA*Macintosh            74 @ 246
6:       Apple_Patches Patch Partition     512 @ 320
7:           Apple_HFS untitled        2048000 @ 832     (1000.0M)
8:           Apple_HFS untitled 2       102400 @ 2048832 ( 50.0M)
9:           Apple_HFS untitled 3      2082358 @ 2151232 (1016.8M)
10:          Apple_Free Extra               10 @ 4233590

Device block size=512, Number of Blocks=4233599 (2.0G)
DeviceType=0x0, DeviceId=0x0
Drivers-
1: @ 64 for 22, type=0x1
2: @ 118 for 36, type=0xffff
3: @ 192 for 20, type=0x701
4: @ 246 for 33, type=0xf8ff
```

If you created your partitions originally in Drive Setup as I did, you can delete each partition you want to convert individually, and use the block base and block length values to create new A/UX partitions. Here are a few useful definitions:

- block = 512 bytes (typically) - the actual block size will be specified when you print your partition map
- block base = block number where that partition starts. This can be entered in terms of the actual block or by the partition # followed by the letter 'p', i.e. 704 or 5p for partition #5
- block length = length of the partition in blocks. This can be entered in terms of blocks or by the size of the partition in terms of kilobytes, megabytes, or gigabytes, followed by the letter k, m, or g, respectively. Example: partition #9 would be entered as 102400 or 50m. This entry must be a whole number - for example, if you wanted to create a 0.5 GB partition you would use 512m, as 0.5g would give an error.

On my partition map, the first HFS partition that I want to convert is root, located at partition #7, which starts at block 832 and has a length of 204800. To convert it, I would first delete the old HFS partition where it's at, and then create a new A/UX partition in its place. Here is an example of the commands I would use (my input is in red):

```
Command (? For help): d 7
Command (? For help): n
First block: 832 (or 7p)
Length in blocks: 204800 (or 1000m)
Name of partition: root
```

I would then print my partition map (p) again to check if it was correct. Next, I would perform the same steps for swap at partition 8, opt at partition 7, home at partition 8, and swap at partition 9 to convert these to A/UX.

My finished partition map looks like this.

```
Command (? for help): p
```

```
Partition map (with 512 byte blocks) on '/dev/hde'
 #:                type name             length   base    ( size )
 1: Apple_partition_map Apple                63 @ 1
 2:      Apple_Driver43*Macintosh            54 @ 64
 3:      Apple_Driver43*Macintosh            74 @ 118
 4:    Apple_Driver_ATA*Macintosh            54 @ 192
 5:    Apple_Driver_ATA*Macintosh            74 @ 246
 6:       Apple_Patches Patch Partition     512 @ 320
 7:     Apple_UNIX_SVR2 root            2048000 @ 832     (1000.0M)
 8:     Apple_UNIX_SVR2 swap             102400 @ 2048832 ( 50.0M)
 9:           Apple_HFS untitled 3      2082358 @ 2151232 (1016.8M)
10:          Apple_Free Extra                10 @ 4233590


Device block size=512, Number of Blocks=4233599 (2.0G)
DeviceType=0x0, DeviceId=0x0
Drivers-
1: @ 64 for 22, type=0x1
2: @ 118 for 36, type=0xffff
3: @ 192 for 20, type=0x701
4: @ 246 for 33, type=0xf8ff
```

After checking that it's correct, I would then use the write command (w) to save the partition map, and then quit pdisk (q). If you mess up, just quit without writing the map, and go back into pdisk from the installer and convert your partitions again. After you quit, you will be back on the Partition Disks screen. Tab over to "done" and press return.

Current Disk Partitions Screen
This screen allows you to choose the mount points of each of your partitions. You must at least include your root partition at "/". Use the up and down arrows to select the partition, then tab to Edit and hit space to select. A new window then pops up that allows you to edit that partition. Enter the mount point (i.e. /) and then tab to OK. If you have any other partitions, such as /usr, add the mount points for these partitions as well. When finished, tab to OK and hit space to continue.

Active Swap Space Screen
Select the partition you created for swap and select "Ok" - it will take a little while to format the partition.

Select Partition screen (for installation from local hard drive only)
Choose the Mac OS drive where you created the Red Hat folder (with the base and RPMS folders). For the "Directory holding Red Hat:" box, enter "/software" (without the quotes). Select Okay - the installer will then scan your RPM packages and check for missing files by comparing them to the list in the comps file. If you get any comps errors, write them down so that you can install those packages later individually. If you get a lot or if those files are critical for booting, you might even want to reboot into Mac OS and download them and put them in the RPMS folder before you continue. You'll have to start over in the installer, but you can skip the pdisk part. If it's a package you know you don't need and nothing else depends on it, just ignore the warning. *Note:* older versions of comps will report an error for Xautoconf – this is due to an error in the comps file. You will need to manually install Xautoconfig yourself after the installation. Alternatively, you can install the updated comps file from ftp://ftp.linuxppc.org/linuxppc-1999/RedHat/base/comps.

Partition to Format Screen
You should format all partitions that you would like to use for LinuxPPC.

Components to Install Screen
In this screen, you can either choose entire groups of packages or further customize your installation by selecting individual packages. You can find out more about each package by pressing F1 or by visiting http://dev.linuxppc.org/RPMS/. After you have selected the appropriate packages, it will then warn you

about unresolved dependencies. This basically means that one or more of the packages you have chosen to install requires files or libraries installed by another . You should select the option to "Install packages to satisfy dependencies", then select OK.

It will then go through the actual installation of files – this can vary from just a few minutes to over half an hour, depending on the number of packages you have selected and the speed of your machine.

Mouse
The installer will then probe for your mouse. This can take a few minutes. Afterward, you will be given an opportunity to configure your mouse. If it is a one-button mouse, you will want to select the "emulate 3 buttons" option. You can later specify which keys will be used for emulation (more about this in the next chapter on booting LinuxPPC and in the User Guide).

Network Setup
The installer then asks about configuring LAN networking. You can choose to configure your network for a static IP, BOOTP, or DHCP.

Timezone
When setting the Timezone, note that Macintosh internal clocks do not store time in GMT.

Services Setup Screen
In this section you can select which administrative and network services you want to begin at bootup. You can use the F1 key over each service to get more information.

Configure Printer Screen
You can now configure your printer at this screen. All postscript network printers and some Apple Stylewriters, Hewlett Packard printers, and Epson printers are supported (the User Guide will be updated later to include which printers are supported and information on how to set them up).

Root Password
For your root password, make sure you don't use a common word in the dictionary, as the root password is perhaps the most critical aspect of system security.

Authentication Configuration
Most users can leave these at the default values.

BootX Configuration
Write down the settings that the installer gives you for use with BootX.

Video Probe
The installer will then probe for the graphics card or built-in chipset in your system. It will then allow you to set up your monitor and select the video modes that you may use as well as the color depth (8-bit, 16-bit, or 24-bit). Finally, it will start up the X server to test your setup. These settings will be stored in the /etc/X11/XF86Config file in your installation. If the X server fails to start, you can reconfigure it later after you've booted into LinuxPPC – this will be covered in the next chapter.

Booting LinuxPPC

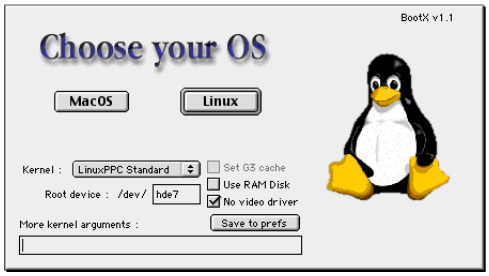
This section describes:

- Setting up BootX
- Logging in
- Getting around the desktop

Setting up BootX

If you used the X Linux Installer:

Under the root device field, enter the partition name that you set as / during installation. Uncheck the "Use RAM Disk" box and clear the "More kernel arguments" field. Keep the "No video drivers" box checked - you can set up the X windows server separately to use accelerated video. Hit tab so that the Linux button is highlighted to make it your default OS, then click on the "Save to prefs" button. Then hit return to boot into LinuxPPC R5.



BootX settings for launching R5 for a system installed using the X installer

If you used the Red Hat Installer:

Under the root device field, enter the settings you were given at the end of the installation. Uncheck the "Use RAM Disk" box and clear the "More kernel arguments" field.

If the X server was successfully configured during installation, you can uncheck the "No video driver" box. ATI video acceleration can also be enabled if your video chipset was identified as ATI Rage II, Rage Pro, or Mach64 during installation: Under the "More kernel arguments" field, you can enter:

```
video=atyfb:vmode:[vmode_setting],cmode:[bit_depth]
```

where [vmode_setting] is the resolution/frequency mode you would like to use, and [bit_depth] is the color bit depth you would like to use.

Vmode	Resolution
1	512x384 60Hz (Interlaced-NTSC
2	512x384 60Hz
3	640x480 50Hz (Interlaced-PAL)
4	640x480 60Hz (Interlaced-NTSC)
5	640x480 60Hz
6	640x480 67Hz
7	640x870 75Hz (Portrait)
8	768x576 50Hz (Interlaced-PAL)
9	800x600 56Hz
10	800x600 60Hz

Vmode	Resolution
11	800x600 72Hz
12	800x600 75Hz
13	832x624 75Hz
14	1024x768 60Hz
15	1024x768 72Hz
16	1024x768 75Hz
17	1024x768 75Hz
18	1152x870 75Hz
19	1280x960 75Hz
20	1280x1024 75Hz

Valid cmode bit depth settings are 8, 15, 16, and 24, which correspond to 256, 32768, 65536, and millions of colors, respectively. You can also set cmode to 32, although this gives the same results as 24.

If the X server was not setup correctly during the Red Hat installation:
If X was not configured successfully during installation, keep the "No video driver" option selected for now and type in the word "single" under the "More kernel arguments" field. This will put you into runlevel 1, which is single user. This prevents the system from automatically booting into X windows when you start up, and drops you directly into a console shell. Click the "Linux" button and wait for the kernel to finish loading.

FYI: The R5 installation will normally boot you into runlevel 5, which is multi-user X windows. This is controlled by settings in the /etc/inittab file, which will be discussed later in the User Guide.

You will eventually get dropped into the bash shell prompt (the number before sh# is the version number):
2.03 sh#

At this point, you should try running Xautoconfig to configure your X server automatically (type in /usr/bin/X11/Xautoconfig at the prompt). It's non-interactive, so it does its thing and exits. The file that it creates/modifies is /etc/X11/XF86Config, which is used by the Xfree86 server. Alternatively, you can use /usr/bin/X11/Xconfigurator to run the configuration tool that was used by the Red Hat installer. This will also edit /etc/X11/XF86Config.

Note for PowerBook G3 Users: If you have a 14.1" LCD screen, you can also try selecting the Apple Studio Display 15" LCD as your monitor in Xconfigurator if you did not have success with the PowerBook G3 selection.

To determine if /etc/X11/XF86Config has been configured correctly, try typing in startx at the prompt to launch an X window session. You can also type in "init 5" to switch to runlevel 5 – this will start the gdm graphical login manager.

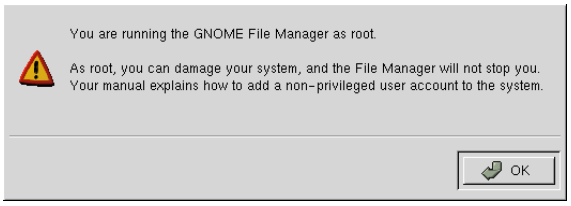
If you are still having problems with configuring and starting X windows, refer to the X windows configuration section in the User Guide.

Finally, if you add "adb_buttons=120,122" at the end of the more kernel arguments field, it will emulate the middle and right mouse clicks with the F1 and F2 keys, respectively (this is only for systems installed by the Red Hat installer – systems with the X installer use option-2 and option-3 as the middle and right mouse clicks). Use tab to select your default operating system, and click on the Save to prefs button. So finally, BootX settings for a Red Hat installed system might look like this:

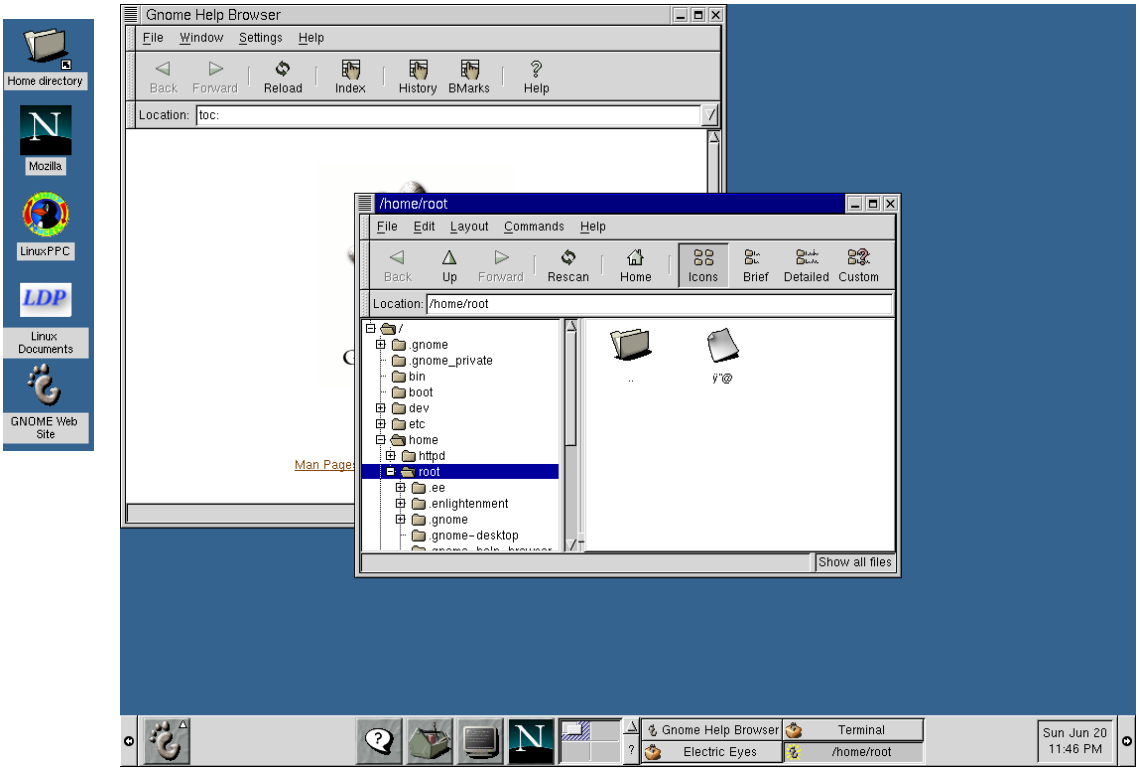
BootX settings for launching R5 from a Red Hat installed system

Logging In
When the system boots, it will automatically load the X Windows system. You will then be presented a log-in screen, where you should log in as root initially and use the password you supplied during installation. Note that you won't see anything as you type in your password – this is a security feature of most Unix-like systems.

You will get an alert about using GNOME while logged in as root - don't worry, nothing is broken - it's just a warning about the power of being superuser, and how you should be careful whenever you are logged in as root. The system won't stop you from doing something dumb, such as deleting critical parts of your system.



You will then be introduced to the GNOME desktop and the Enlightenment window manager. The Gnome desktop is an ongoing effort by GNU (GNU's Not Unix) to create a free desktop package with integrated tools and applications. It supports a variety of window managers, such as Enlightenment, WindowMaker, AfterStep, and fvwm. These window managers provide different graphical user interfaces, and even within each window manager you can use different settings and themes to suit your taste.



The R5 GNOME Desktop

On the left hand side of the desktop is a row of icons (hidden behind the Gnome help browser in the screenshot above). These correspond to your Home directory through the Gnome file manager (the window shown in the middle), and Netscape links to the Mozilla website, LinuxPPC.org, the Linux Documentation Project, and the GNOME web site, respectively. The GNOME help browser provides information from the GNOME user guide in html format, as well as Linux information from "man" pages (short for manual), info pages, and other information on GNOME applications. If you get an error while trying to load the Gnome User Guide, the gnome-userguide rpm may not have been installed properly. We'll cover installing rpms later here in the R5 user guide. The latest revision of the GNOME user guide is always at http://www.gnome.org/users-guide/index.html.

The Gnome file manager provides a graphical user interface for you to browse your directories and files. Gnome controls what to do with different file types using its MIME type (Multipurpose Internet Mail Extension). Thus, when you double click on a .gif file, it will open it in an image utility. Please refer to the Gnome user guide for more information on configuring the file manager and adjusting file associations.

On the bottom of the screen is the Gnome panel. The bottom left button with the footprint icon is the main menu, analogous to the Start menu in Windows or the Apple menu in Mac OS. Going from left to right along the Gnome panel is the help button, Gnome control center button, Xterm launcher button, and a Netscape icon button. Next we have the Gnome pager, which controls the virtual desktops within your Gnome session. In the default setup, there are four virtual desktops. Next there is an arrow pointing up and a button with a "?" mark – this allows you to configure your Gnome pager. Next there is a task list, and finally the Gnome clock. On the far left and right of the panel are arrows that collapse the Gnome bar when clicked.

About Virtual Desktops
Virtual desktops can help you organize your work more effectively and maximize your screen real estate. It's almost like having extra monitors. The current active desktop is shaded, and you can see outlines of where your windows are. To switch to a different virtual desktop, you can either click on the outline of the desktop, or move your mouse to the edge of the screen towards that desktop.

Multiple Mouse Buttons
X Windows was originally designed for systems with three mouse buttons. If your mouse only has a single mouse button, you will need to emulate the other mouse buttons to access certain menus and commands. The default mouse button is the left button. If you used the X Linux Installer, a middle-click can be emulated by option-clicking or holding down the option-2 key combination. The right mouse click can be emulated by holding down the option-3 key combination. If you used the Red Hat installer, the middle and right mouse buttons will be emulated by the F1 and F2 function keys, as specified by the "adb_buttons=122,120" kernel arguments given in BootX. Keycodes 122 and 120 correspond to F1 and F2, respectively. We'll cover how to find different keycodes and assign them later in the User Guide.

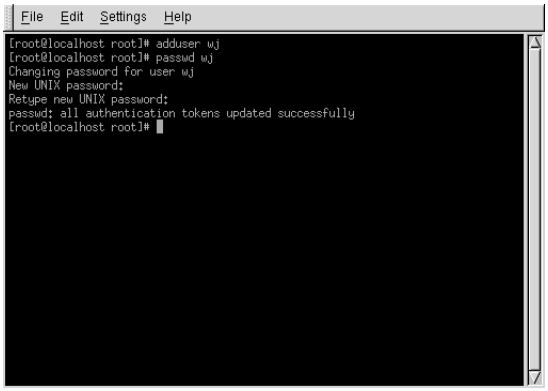
You can get a special pop-up menu for your window manager (Enlightenment in this case) by middle-clicking in the background. You can get a Gnome pop-up menu by right-clicking. Middle-clicking on items on the Gnome panel allows you to select special options for that item.

Opening Up an Xterm Window
To open up an Xterm window, click on this icon on the Gnome bar. 

An Xterm window provides an interface for the bash shell (there are several different types of Unix shells, bash is the default). A shell is a command-line interface where you can run programs, manage files, work with scripts, and much more. It's where the power and flexibility of Unix systems lie, although it does take some time to learn all the commands at your disposal. The User Guide will cover some basic commands to get you started.

Adding Additional Users
The first task we'll try in an Xterm window is adding an additional user to your system. This is performed using the adduser command – its syntax is "adduser [username]". Next, you should assign a password to

this user. This is done using the passwd command – its syntax is "passwd [username]". You would then be prompted to type in the new password twice. See the image below for an example.

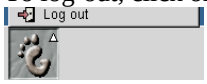


Adding a new user

The passwd command can also be utilized to change your own password. Simply type in "passwd" and hit return. It will then prompt you for your old password, and ask you to type in your new password twice.

Logging Out of the GNOME Session

To log-out, click on the Gnome menu icon on the lower left hand corner, and select logout:



This will bring up a login confirmation screen, where you can save the current setup (any changes that you've made to the gnome environment, such as opening Xterm windows, closing the file manager, etc.). You can also select Logout, Halt (same as shutdown), or Reboot. Note that the Halt and Reboot options will only work if you are logged in as root. If you choose Logout, it will bring you back to the login screen, where you can click on the Options button and select System > Reboot or Halt if you need to restart or shutdown the machine as any user.

Oh, and before I forget… welcome to LinuxPPC.

LinuxPPC User Guide

[under construction]

This section describes:

- Basic Linux Commands
- Installing with rpm
- Setting up PPP
- Mounting another hard drive partition
- Configuring foreign language keyboard
- Setting up netatalk
- Building your own kernel

Basic Linux Commands

You already know that you can open up an Xterm window by clicking on the Xterm icon on the Gnome panel, but you can also open up an Xterm by right-clicking and selecting New > terminal. Both these methods open up Gnome X terminals – there are also several different types of terminals, such as RXVT, color Xterm, and kterm, which are accessible through the Utilities submenu under the Gnome menu.

The first Linux command we'll cover is "man" – this is short for manual, and it's the standard utility used to find out more about Linux commands. More recently, the "info" command is gaining popularity – it's more flexible than man and it also includes all the man definitions. To use the man utility, you type in

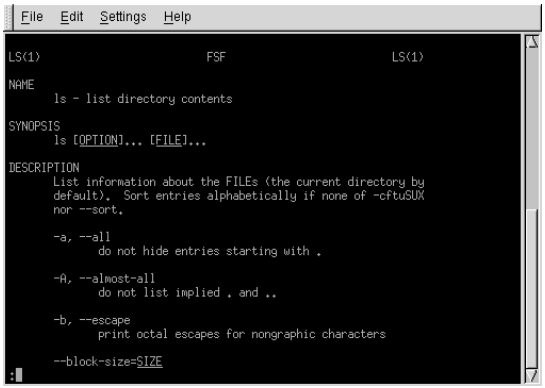
```
man [command]
```

at the prompt, and hit return. You can use the info command using the same syntax: `info [command]`. The info command by itself gives a listing of available info commands. You can also use the Gnome help browser to view various man and info pages.

You can go ahead and use man to find out more about "ls", which lists the contents of a directory.

```
man ls
```

This gives us a screen that looks like this:

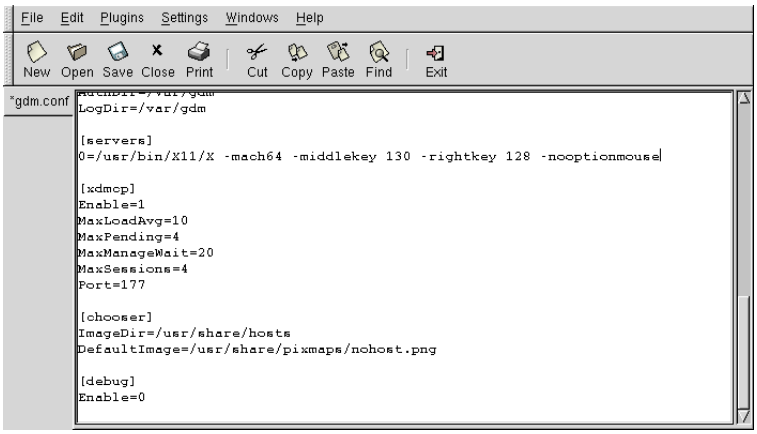


Note that it starts out with the name and a brief description, followed by a synopsis of the syntax on how ls is used, and a description of how it is used and its options.

If you need more information not provided in this guide, refer to the documentation in the `/usr/doc` directory. You can also find valuable information in the FAQ-O-Matic (http://www.dartmouth.edu/cgi-bin/cgiwrap/jonh/lppc/faq.pl), Linux-pmac mailing list archive (http://www.linuxppc.org/archive/), the comp.os.linux.powerpc newsgroup, Linux Documentation Project (http://sunsite.unc.edu/LDP/), and the Red Hat 5.0 Manual (http://www.redhat.com/support/docs/rhl/RHL-5.0-Installation-Guide-HTML/manual/).

Setting up Accelerated X Windows

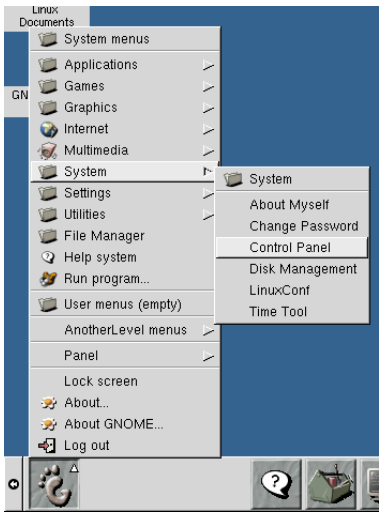
gedit /etc/X11/gdm/gdm.conf – scroll down to bottom to [servers], edit line below to get accelerated video on ATI Rage II and Rage Pro graphics by adding –mach64. Twin Turbo 128 users would use –tt128. The –middlekey and –rightkey modifiers substitute F1 and F2 for option-2 and option-3 for the middle and right mouse buttons, -nooptionmouse eliminates the need to press option down while emulating these mouse clicks.



Sharing Files with Mac OS

Only used with HFS formatted drives, can mount drives using the following command:
mount /dev/[drive partition] /mnt
Files then accessible from /mnt directory.

Wrapping Up



If you are still having problems using X windows, reboot the machine and go back into Mac OS. The 2.3.6 and later kernels include new code for ATI graphics chipsets to autodetect the system's memory clock settings – this is particularly useful for Powerbook users, who are limited by their fixed-frequency LCD displays. However, R5 ships with an earlier 2.2.6 kernel, so all the system modules and system map are for that kernel revision. You can download a 2.2.6 kernel that includes the ATI code from 2.3.6 at http://www.linuxppc.org/userguide/vmlinux_fb.sit. Decompress it with a program such as Stuffit Expander and place it inside the Linux Kernels folder located inside the System Folder, and select it in BootX. Then enter the video mode and color mode that you'd like to use (refer to the section on using BootX with Red Hat installed systems in the previous chapter).

Installing with rpm

To install with rpm, download the rpm package you want to install (until you can get ppp or other networking to work, you'll have to use Mac OS, BeOS, AIX, etc.). If you placed the rpm file in the root level of your Mac OS partition and use /mnt as your mounting point, you can issue the following command to install or update rpm packages:

```
rpm -Uvh /mnt/(name of rpm file)
```

Note: since the Mac OS has a 31 character filename limit, some of the last few letters of the rpm file name might be cut off. Rpm will still recognize it as a rpm file even if the .rpm extension is missing - just make sure you type in the filename as its listed in the directory.

On some files, you may get errors that the package has already been installed or that it cannot be installed due to problems with dependencies with other packages. If you're sure there are no problems and would like to override the error, you can use

```
rpm -Uvh --force --nodeps /mnt/(name of rpm file)
```

To install over ftp, use

```
rpm -Uvh ftp://ftp.linuxppc.org/pub/linuxppc/linuxppc-R4/RedHat/RPMS/(name of rpm file)
```

Setting up PPP

If you have KDE installed, then using kppp is definitely the way to go. Just make sure that /dev/modem is linked to /dev/cua0 if it's on the serial port or to /dev/cua1 if it's on the printer port:

```
ln -s /dev/cua0 /dev/modem.
```

Otherwise, I think the easiest way to get ppp running on your machine is to use ezPPP - there is a rpm of it at MkArchive (ftp://ftp.sunet.se/pub4/os/Linux/mklinux/mkarchive/comm/connecting/ezppp-1.0B9-1.ppc.rpm). Take a look at the documentation at the ezPPP home page http://www.serv.net/~cameron/ezppp/howto.html.

You can also try netcfg, which is a convenient X utility to set up networking, including ppp. It's an rpm package that is installed in a standard installation. If you have problems using netcfg, documentation on using using it is available at http://nitro.med.uc.edu/DR3web/netcfg.html.

Mounting another hard drive partition or other media

To mount another hard drive that isn't mounted at start-up, you would use the mount command. Its format is usually `mount [device] [mounting point]`. Mounting point is any empty directory that will serve as the location where you can access the mounted partition. You usually have to pass special flags such as the partition type using the `-t` flag (`hfs` or `ext2`) and whether or not the partition will be read-only (`-r`).

```
mount -nr /dev/hdaY /mnt   allows you to mount a ide partition #Y in read only mode
```

```
                          (remove -r to go into read/write).  When mounting HFS
                          partitions, it's probably safest to use read only so
                          you don't mess up the Mac's b-trees.
                          You'll have to mkdir /mnt before you can mount anything to it.
                          /mnt is an arbitrary directory name - you can use another name
                          if you wish

mount -nr /dev/sdXY /mnt  same as above, but for scsi (X is the device ID, usually a
                          letter between a and f that the kernel gives for that drive)

mount -wn -o remount /dev/hdaY    remounts your root directory as read/write in case
                                  something screwed up and it mounted only in read mode.
```

I generally use `mount -t (type) /dev/…` to access drives. For example, to access a floppy I'd use (assuming I've made a /floppy dir):

```
mount -t hfs /dev/fd0 /floppy
```

For another unix partition (such as a ZIP drive): `mount -t ext2 /dev/sda5 /zip`

To get partitions to mount during startup, you need to add the device name to `/etc/fstab`. Follow the format for the other partitions already listed there.

Foreign Keyboards

Henrik Linder reports that to get a Swedish keyboard working in LinuxPPC, use the instructions at http://www.dna.lth.se/~roger/MkLinuxKeyboard.html.

For a French keyboard, Stephane Boissan reports that the instructions at http://www.inforoute.capway.com/le2/post-install.html work well.

Netatalk

Netatalk allows you to set up print and file server using AppleShare and AppleTalk. Information on setting up netatalk is available at http://www.linuxppc.org/userguide/configure/netatalk/.

Building your own kernel

To build your own kernel, you'll need the latest kernel sources. For Powermac/clones, the official kernel sources are in Australia at ftp://samba.anu.edu.au/linux-pmac/kernel-source. It is also mirrored in the US at ftp://ftp.linuxppc.org/pub/mirrors/pmac/kernel-source. Most users will want the 2.1 kernel. Download the latest `linux-pmac-XX.tar.gz` in binary (around 6 MB). If you don't have them installed yet, you will also need the C development and development libraries rpm packages from ftp://ftp.linuxppc.org/pub/linuxppc/linuxppc-R4/RedHat/RPMS/ or any of its mirrors. Install the rpm packages. Then put the kernel source in `/usr/src/`. Next, remove any old linux and linux-pmac directories and install the kernel sources:

```
cd /usr/src
rm -rf linux linux-pmac
tar -xvpzf linux-pmac*.tar.gz
```

Next, create a symbolic link linux to point to linux-pmac

```
ln -s /usr/src/linux-pmac linux
```

Go to the `/usr/include` directory and do some cleaning up and make sure everything is linked properly:

```
cd /usr/include
rm -rf asm linux scsi
ln -s /usr/src/linux-pmac/include/asm-ppc asm
```

```
ln -s /usr/src/linux-pmac/include/linux linux
ln -s /usr/src/linux-pmac/include/scsi scsi
```

Finally, clean up old .o files and dependencies

```
cd /usr/src/linux-pmac
make mrproper
```

Next you can configure your kernel using `make config`. This can take quite a while if you're new to it, but most of the defaults should be fine until you get used to it. You can press return to accept the default or press ? to get more info on the item.

Do a `make dep` to set up dependecies followed by a `make clean`

Finally, do a `make vmlinux` to create a regular kernel or make `vmlinux.coff` to create a floppy-bootable version. On my StarMax 4000 (604e 160 MHz, 96MB ram) it takes 10-15 minutes. Vmlinux will be placed in `/usr/src/linux-pmac` while vmlinux.coff will be placed in `/usr/src/linux-pmac/arch/ppc/pmac/boot/`.

When the kernel sources are updated, there is no need to download the entire kernel source again. You can simply download the patches dated in between the source you have and the newest source. These are usually named `patchXX.gz` and are found in the same directories as the kernel sources. You will have to apply the patches in chronological order, from oldest to newest. Don't skip any. To apply patches, you can use:

```
cd /usr/src
gzip -cd patchXX.gz | patch -p0
```

For more information on building a kernel, see the README that's in `/usr/src/linux-pmac/`. Also look at some of the resources at the Linux Documentation Project. (http://sunsite.unc.edu/LDP/)

Other useful commands

If you're new to LinuxPPC or Linux in general, some of these commands might be useful:
(anything in parentheses are arbitrary names)

```
tar -xvpzf (tar_archive.tar.gz)     allows you to unzip and untar a file (remove the z
                                    flag if it's not gzipped)
du -s [directory]                   lists the size the directory
ln -s (source) (alias)              creates an alias to link to another directory or file
chmod +x (file)                     changes a file to executable
find [directory] [filename/dir name]  finds the path of a file or directory from a parent directory
vmode X Y                           switches the video resolution and bit depth (see man pages)
startx                              begins an X windows session
sndvolmix [volume level]            adjusts volume level from 0 to 9.  Using the -c flag
                                    toggles cd playthrough you can use any of the various
                                    cd players to play audio cd's, such as kde's kscd
ps                                  lists current processes by PID
top                                 shows information on ram and cpu usage by process
kill (pid)                          kills the process with that particular PID
shutdown -h now                     shuts down the system
shutdown -r now (or reboot)         reboots the system
man (command)                       gives information on that command (you can also try
                                    (command) --help)
```

There are countless other commands, but those should get you started.

Links & Reference

[under construction]

Mirrors for the LinuxPPC distribution

Latest kernels:

- ftp://samba.anu.edu.au/pub/linux-pmac (Australia), official site. (be nice to Australia - use a mirror if you're in US!)
- ftp://ftp.dodds.net/pub/linux/pmac (USA), daily updated mirror
- ftp://ftp.linuxppc.org/pub/mirrors/pmac

Make sure you download these in BINARY mode! For Mac OS users, Fetch is recommended. To download in binary form in Netscape, hold down the option key while clicking the link. If that doesn't work, hold down the button while you click on it until you get the pop-up window. Select "Save this Link as.." and when it pops up the save dialog, choose to save as "Source" format. To avoid problems, Netscape and IE users should configure their browsers so that file types with .rpm, .coff, .tgz. and .gz are downloaded in binary. In Netscape, this is in the Preferences > Navigator > Applications setting. In IE, this is in the Preferences > Receiving Files > File Helpers setting.

RPM files

Here are some sources of .ppc.rpm files that you can install:

- ftp://ftp.linuxppc.org/pub/linuxppc/linuxppc-R4/RedHat/RPMS directory
- ftp://ftp.linuxppc.org/pub/linuxppc/contrib/linuxppc-R4/RPMS/ directory
- MkArchive - ftp://ftp.sunet.se/pub4/os/Linux/mklinux/mkarchive/index.html

Here are a few Mac OS utilities that may come in useful in your adventures with LinuxPPC:

· Stuffit Expander to unbinhex and unstuff files (free, at http:// hyperarchive.lcs.mit.edu/Hyperarchive).

· MacGzip to decode gzipped files (free, at http:// hyperarchive.lcs.mit.edu/Hyperarchive)

· Boot Variables - a must have! It allows you to configure your Open Firmware boot variables so that it knows what to bootup. You can set it to boot MacOS or Linux-pmac via a floppy or off a hard drive, among other things. It also lets you save configurations, so that you can have one configuration you can use to boot MacOS and another for Linux-pmac and don't need set the variables manually on bootup (free, from ftp://samba.anu.edu.au/pub/linux-pmac/ or ftp://ftp.linuxppc.org/pub/linuxppc/linuxppc-R4/RedHat/tools/).

· Quik-MacOS - useful for installing new kernels (vmlinux) and installing/configuring the bootstraps for booting (free, from ftp://samba.anu.edu.au/pub/linux-pmac/ or ftp://ftp.linuxppc.org/pub/linuxppc/linuxppc-R4/RedHat/tools).

· LinuxDisks - it won't work on IDE drives, but if you're using SCSI it's pretty nifty to use it to copy stuff from MacOS into your linux partitions. ($15 shareware, available at http://w3.teaser.fr/~mpollet/LinuxDisks/).

· MountX - a Mac OS extension that allows you to mount Linux ext2 partitions in Mac OS (freeware, http://calvaweb.calvacom.fr/bh40/).

· Suntar - a utility that can decode or archive files in the tar format. (free, at http://hyperarchive.lcs.mit.edu/Hyperarchive)

· Tar is a bit out-of-date, but it usually works. (free, at http:// hyperarchive.lcs.mit.edu/Hyperarchive)

- pdisk for Mac OS- It can partition both IDE and SCSI drives into A/UX or HFS types; however, it won't install the Mac drivers if you're looking to use your drive for both Mac OS and Linux. A little tricky to use if you're new to it, but once you figure out the commands then it's not bad. Documentation for it is available at ftp://ftp.linuxppc.org/pub/linuxppc/linuxppc-R4/RedHat/tools/pdisk.html. Its interface is basically the same as the linux pdisk version, so if you have a spare Zip disk and aren't comfortable with partitioning, you can practice with that. (free, originally from MkLinux)

- Apple_HD_SC_Setup_7.3.5 - it doesn't work at all for IDE, and if you want to partition a non-apple drive, you'll have to use ResEdit to change the wfwr resource from 00 to FF (free, from info.support.apple.com).

- Apple Drive Setup - it can't do A/UX partitions, but it can partition both SCSI and IDE drives into HFS partitions. You can change these partitions to A/UX using pdisk in Linux (free, from ftp://info.support.apple.com)

References

If you need more help, you can refer to the following sources:

- FAQ-O-Matic - http://www.dartmouth.edu/cgi-bin/cgiwrap/jonh/lppc/faq.pl -the FAQ-O-Matic is a site by Jon Howell where users have posted answers to commonly asked questions regarding LinuxPPC, Linux-pmac, and MkLinux. It is possible for anyone to register and post answers themselves.
- The Linux Documentation Project - http://sunsite.unc.edu/LDP/ - this is an effort by the entire Linux community to provide documentation about Linux. It includes FAQs and several how-to's.
- http://www.linux.org - the official home of Linux

Appendix A: Open Firmware & Boot Variables

Open Firmware (OF) is used to build the device tree and boot the operating system. Open Firmware boot variables can be modified in three ways:

- with the Boot Variables program in Mac OS (Power Macintosh G3's should use Multibooter)
- with setenv within Open Firmware
- with nvsetenv within LinuxPPC.

The official Open Firmware website is at http://playground.sun.com/1275/home.html. Apple technotes 1061, 1062, and 1063 may also be useful. Also see the /Linuxppc/OpenFirmware category at the FAQ-O-Matic.

To get into Open Firmware, you will need to change some of your boot variable settings. From Mac OS, this can be done with the Boot Variables application. Boot Variables does not work on Powerbook 2400 and 3400. However, this does not matter in these machines since you can go directly into OF at boot-up by holding down the cmd-option-o-f key combination at startup until OF's white screen appears.

Those machines that can run Rhapsody can use Multibooter instead of Boot Variables. It's available from ftp://ftp.apple.com/devworld/Rhapsody/UsefulStuff, and is actually preferable to Boot Variables in Apple G3 machines since it applies some patches to OF.

You can interface directly with Open Firmware using two basic methods:

- with your monitor and keyboard - viewing Open Firmware on your monitor is only possible on versions of Open Firmware which are able to initialize the video based on settings in the output-device boot variable

  To interact with Open Firmware with your keyboard and monitor, set input-device to kbd, output-device to the setting for your machine (if available). Set auto-boot to false. Then write and reboot - you'll get into Open Firmware's interface, which is in Forth. Remember that only these machines with a properly configured, supported output-device can view OF output on their monitor. Refer Appendix B: for output-device settings for different machines.

- with a second computer, a serial cable, and a terminal program like zterm (available at download from hyperarchive). This should work on all models which have an available serial port.

  To talk to OF from a second computer using Zterm, you will have to set input-device and output-device to ttya if you're using the modem port (ttyb for the printer port). Connect the serial cable from the respective serial port to the terminal machine (machine with Zterm).

  Launch Zterm in the terminal machine and change connection settings so that Data Rate = 38400, Data Bits = 8, Parity = none, and Stop Bits = 1. Disable auto-boot in the host machine and then write and reboot - you should soon see Open Firmware output in Zterm. [thanks to Seth Paskin for sending me a guide on this method]

To get into Open Firmware while auto-boot is enabled, reboot the machine and hold down the cmd-option-o-f key combination before the startup chime until the OF screen appears.

To get back into Mac OS, you can enter `bye` at the OF prompt and it will begin booting Mac OS. You can also reset your boot variable settings by zapping your parameter ram (PRAM) by holding down the command-option-P-R key combination during and after the start-up chime. This will also reset many of the settings in your control panels in Mac OS.

Some useful commands in OF (remember that it is case sensitive):

```
devalias                          lists the aliases you can use for your devices
dev / ls                          prints out your device tree
dev [device name] .properties     lists some data about a certain device
words                             lists "commands" for the current device
boot [device]                     tries to boot off that device
bye                               boots you into Mac OS
printenv                          prints your nvram settings
setenv [variable] [setting]       sets your nvram settings
set-defaults                      reset Open Firmware settings
eject fd                          ejects floppy disk
reset-all                         reboots the computer
shut-down                         shuts down the system
```

Useful Open Firmware boot variable options

Variable	Setting
auto-boot?	True / false true makes OF boot the boot-device automatically. False drops you into OF's interface during boot-up (requires setting the input-device and output-device) Alternatively, you can leave auto-boot? True and hold down the cmd-option-o-f key combination to get dropped into OF's interface. On the Powerbook G3 Series, you might have to use the right option key to get this to work.
boot-device	Defines what device you will boot off of. Includes Mac OS ROM, SCSI, IDE, and floppy
boot-file	May be used to define what kernel to use when booting, or pass specific options to the kernel such as the location of root and the video mode resolution for Linux to use. The format for vmode would be vmode=X, where X is the vmode setting. Examples include: vmode=5 (640x480 @ 60 Hz) vmode=6 (640x480 @ 67 Hz) vmode=10 (800x600 @ 60 Hz) vmode=11 (800x600 @ 72 Hz) vmode=12 (800x600 @ 75 Hz) vmode=13 (832x624 @ 75 Hz) vmode=14 (1024x768 @ 60 Hz) vmode=15 (1024x768 @ 72 Hz) vmode=16 (1024x768 @ 75 Hz) vmode=18 (1152x870 @ 75 Hz) vmode=19 (1280x960 @ 75 Hz) vmode=20 (1280x1024 @ 75 Hz) * note that your video chipset may support some modes differently
input-device	kbd, ttya, or ttyb This allows you to input data to OF using the keyboard, modem port (with a 2nd computer), or printer port (using a 2nd computer) It is usually a good idea to set this so that you can input commands to OF, even if you can't see its output.
Output-device	ttya, ttyb, or the OF address for your video chipset on your machine, if supported This only needs to be set if you want to use OF's interface Refer to Appendix B for a listing of each machine.
Load-base	For most machines you should leave it at its default value. For the Apple 5400, 5500, 6360, 6400, 6500, 20th Anniversary(?), Umax C500 and C600, and Power Computing Powerbase you will need to set it to 100000. For Apple G3 computers (desktops, minitowers, and PB G3 Series), it may be necessary to set load-base to 1000000 to prevent problems with gzip. On some G3 desktops/minitowers, you may find it necessary to try setting load-base to `4000`, `60000`, `100000`, or `1000000` to successfully boot/install. Note that load-base cannot be set with Multibooter, so run Multibooter first if you need to, then Boot Variables to set load-base (or use setenv in OF).
Boot-command	set to whatever special command OF should get upon bootup (usually boot). If you're using a SCSI drive and you're getting a lot of DEFAULT CATCH errors during bootup, this may be due to OF trying to boot off the drive before it is ready. Add begin [’] boot catch 1000 ms cr again to your boot-command so that OF will repeatedly try booting the SCSI drive until it works You can also do things such as create a boot-up menu (Appendix E)

Appendix B: Open Firmware output-device Settings

Some models do not support monitor output from Open Firmware, so if you need to see OF's output you'll need to use a second computer and a terminal program, as described in Appendix A: Open Firmware and Boot Variables.

Make sure that output-device is not set to ttya if you have a modem connected to it, as modems sometimes interfere with Open Firmware (the same is true for ttyb if your modem is on the printer port)

For machines with add-on pci cards, you can try the following:
- ATI Xclaim 3d: /bandit/ATY,XCLAIM3D or pci1/ATY,XCLAIM3D or pci2/ATY,XCLAIM3d
- ixMicro TwinTurbo: /bandit/IMS,tt128mb or pci1/IMS,tt128mb or pci2/IMS,tt128mb (use pci1 and pci2 only if your machine has 2 pci buses, such as PM 9500)

Company	Computer	output-device
Apple	20th Anniversary	OF does not sync with built-in monitor
	PB 2400	default settings
	PB 3400	default settings
	PB G3 (original)	default settings
	PB G3 Series	default settings
	4400.00	screen
	5400.00	screen (works for floppy boot only - for scsi/IDE, set to ttyb)
	5500.00	/bandit/ATY,264GT-B*
	6360, 6400	screen (works for floppy boot only - for scsi/IDE, set to ttyb)
	6500.00	/bandit/ATY,264GT-B
	7200.00	no screen output available
	7300, 7500, 7600	/chaos/control**
	8200.00	no screen output available ?
	8500.00	/chaos/control** or /bandit/ATY,XCLAIM or /bandit/ATY,264VT
	8600.00	/chaos/control** or /bandit/ATY,XCLAIM or /bandit/ATY,264VT
	9500.00	/chaos/control** or /bandit/ATY,XCLAIM or /bandit/ATY,264VT or pci2/ATY,mach64
	9600 (ATI)	/chaos/control** or /bandit/ATY,XCLAIM or /bandit/ATY,264VT
	9600 (twin turbo)	/bandit@F4000000/IMS,tt128mb8A or /bandit@F2000000/IMS,tt128mb8A
	G3 (ATI RageII)	screen or /pci/ATY,XCLAIMVR_II or /pci/ATY,mach64_3DU
	G3 (Rage Pro)	screen or /pci/ATY,XCLAIMVRPro or /pci/ATY,mach64_3DUPro
Motorola (and clones)	3000, 4000	screen
	5000, 5500 (ATI)	/bandit/ATY,RAGEII_M or /bandit/ATY,264GT
	5000, 5500 (Twin Turbo)	/bandit/IMS,tt128mb ?
Power Computing	Powerbase, Powercenter Pro	/bandit/ATY,mach64_3D_pcc or /bandit/ATY,264gt
	Powercenter	/bandit/ATY,264gt (some early models can't initialize the screen)
	Powercurve	?
	Powertower, Powertower Pro	pci1/IMS,tt128mb8 or pci2/IMS,tt128mb8
	Powerwave	/bandit/ATY,XCLAIM (200 MHz model may use /bandit/ATY,Mach64 (?)

Umax	Apus 2000	screen
	C500	no screen output available
	C600	no screen output available
	J700	/bandit/IMS,tt128mb
	S900	/bandit/IMS,tt128mb8 or /bandit/IMS,tt128mb

*- for the Powermac 5500, OF display may appear dim You can try using the following in the boot-command to brighten it up (thanks, Jens Ch. Restemeier)

```
0f3000030 000ffcf76 over ! 4 + 0f02f7f7a over ! 4 + 072ff3e00 swap ! boot
```

** - OF initialization of video using /chaos/control/ is inconsistent (OF version 1.0.5). Alan Mims sent in a fix for OF 1.0.5 to the Linux-pmac mailing list, which is in Appendix C.

Appendix C: Fixes for Apple OpenFirmware 1.0.5

[from the linux-pmac mailing list]

Alan Mimms
Sun, 17 Aug 1997 19:08:19 +0000

Hi.

I have finally had enough with the buggy screen and disk drivers in Apple's Open Firmware on PowerMac 7200, 7500, 7600, 8500, 9500, and probably some others I'm forgetting. So I have written some NVRAMRC based patches to Open Firmware to hack around the bugs enough for my purposes anyway. I hope the hacks may help you too.

Start up and break into your Open Firmware -- Cmd-Opt-O-F during the boot beep held down until the "user interface" for Open Firmware comes up on your screen or, if you're smart, your serial port. The banner printed by Open Firmware shows the Open Firmware version. These patches ONLY apply to Open Firmware version 1.0.5. Other versions will crash if these patches can be applied at all!

Type at the prompt:

```
nvedit
```

and hit control-L to see the cryptic stuff that is part of your Open Firmware's startup sequence. This stuff patches various bugs in the ROM.

Hit control-N enough times that you no longer see a new line of gobbledegook every time several times in a row. This means you're at the bottom. Either paste the following into a terminal session (NOTE: must be at least a dozen or so ms delay between characters and maybe 100ms between lines to work right!) or else enter the following lines very very carefully:

```
dev /bandit/gc/via-cuda
' write value &W
: -&We &W swap - execute ;
: P1 4D8 -&We false 548 -&We ;
&W FC + ' P1 Blpatch
: P2 0C 2 ms ;
&W E0 + ' P2 Blpatch device-end

: wBoot
begin
 boot-device ['] $boot catch drop
 ." -Waiting for boot-device" cr
 d# 500 ms
key? Until
;
```

Note that ALL whitespace above except for the line indentation is REQUIRED. FORTH is a very very very strange "language". It may be safe to leave the line indentation as I have it above when pasting if you wish. It's wasteful but who cares?

At the end of this laborious typing (or pasting) session hit control-C (yes

that's right: control-C is the end of editing session character in Open Firmware). Then type at the prompt

```
nvstore
```

to save the changed NVRAMRC variable into NVRAM.

The first block fixes a bug in the via-cuda driver in which not enough time is given for the device to settle when it is told to set the video controller's clocks up.

The second block defines a FORTH word that can be used in place of the normal boot-command contents to wait for the disk to spin up before attempting to really boot. This avoids the standard "black screen the first time you power on the computer each day" problem.

Then type at the prompt

```
setenv boot-command wBoot
```

This sets up the default command executed on auto-boot (normally on power on) to run the above disk spinup waiting hack.

FYI: the reason I know about this stuff is that I worked for about two years as the Copland booting guy. Sheesh…

Please don't inundate me with a bazillion questions about Open Firmware. I have a real job and it takes 60+ hours a week of my time. I get PAID to do it. I just did this to fix MY PPC Linux box and I wanted to help you folks out a teeny bit if I could. No, I don't work at Apple anymore. Cancelling Copland was the last straw. I worked there more than nine years. (sigh)

Happy trails.

Appendix D: Partitioning with pdisk and fdisk

The Red Hat Installer includes pdisk for you to partition drives (the installer refers to using fdisk, but it is actually linked to a version of pdisk). It is capable of partitioning both SCSI and IDE drives and should work with drives from any manufacturer. The only difference between using fdisk (pdisk) in the installer and pdisk in Mac OS or within Linux is the syntax that each uses for its commands.

The version of pdisk that runs as a Mac OS application was ported by the MkLinux team and also uses a command line interface. They have provided documentation on using this version of pdisk. These can be found at ftp://ftp.linuxppc.org/pub/linuxppc/linuxppc-R4/RedHat/tools/. Note that the partition naming scheme that they use (/dev/scsi0.1, /dev/ata0.0) is different than what LinuxPPC uses or what Open Firmware uses.

Each hard disk requires a partition map as the first partition. Disks used for MacOS also require partitions which contain drivers and other patches. The rest of the partition map consists of the hard drive partitions - typically, this will initially be one large Apple_HFS partition dedicated to MacOS.

Once partitions have been created, they are independent of any other partitions on the same disk. Each partition can be changed to a different type or reformatted without affecting any other partitions sharing the same drive. Unfortunately, splitting partitions will not protect them from hard drive failure.

In general, a Linux user coming from Mac OS will be concerned with two partition types here: Apple_HFS for MacOS, and Apple_UNIX_SVR2 for Linux-pmac. While fdisk/pdisk can create Apple_HFS partitions, it cannot write the proper Mac OS drivers so that they can be used for MacOS. It also will not format the partitions. Mac OS applications such as Disk First Aid, HD setup, and Drive Setup can properly initialize these partitions once they have been created in fdisk/pdisk. These partitions can later be formatted into HFS or HFS+ by the Mac OS.

Using fdisk/pdisk

reminder: partitioning usually results in making all old data on your drive inaccessible, so remember to back-up first

The Red Hat installer will drop you directly into fdisk (pdisk) after you select which disk to partition. Otherwise, if you're calling up pdisk from a shell prompt, you'd use the syntax `pdisk /dev/hdX` (where X is a, b, c, etc.) if you're editing an IDE drive or `pdisk /dev/sdX` for a SCSI drive.

Fdisk and pdisk commands		
fdisk	**pdisk**	**function**
?	?	help
p	p	prints the partition table
I	I	initialize the partition map (erases the old map)
n	c	create a new Apple_UNIX_SVR2 partition
N	C	create a new user-defined partition type (Apple_UNIX_SVR2 or Apple_HFS)
d	d	delete a partition (turns it into "Extra" space)
w	w	write the new partition table (permanently saves changes)
q	q	quit (does not automatically save changes)

The first thing to do when you start-up fdisk/pdisk is to print your partition map (with p). It will look something like this:

```
/dev/hda
   #:                 type name                    length   base       ( size )
   1:  Apple_partition_map Apple                       63 @ 1
   2:       Apple_Driver43 Macintosh                   64 @ 64
   3:     Apple_Driver_ATA Macintosh                   64 @ 128
   4:        Apple_Patches Patch Partition            512 @ 192
   5:            Apple_HFS untitled               5008048 @ 704        (  2.4G)

Block size=512, Number of Blocks=5008751
DeviceType=0x0, DeviceId=0x0
Drivers-
1: @ 64 for 18, type=0x1
2: @ 128 for 30, type=0x701
```

The numbers under the # sign are the partition numbers. The length is the size of the partition in blocks (typically, 1 block = 512 bytes), and the base is the block number where the partition starts. Since each block is 512 bytes, it takes 2048 blocks to make up one megabyte. Take partition #5 for example - its length is 5008048: dividing by 2048 gives you 2445.3 MB, shown in the "size" column as 2.4G.

Partitions can only be created within free space (type = Apple_Free). These are made by deleting existing partitions, using the `d [partition #]` command. Once you have free space, you can create new partitions within it using the `c` or `C` commands in pdisk (`n` or `N` commands in fdisk). Using `C/N` requires that the user specify what type of partition is being created; otherwise using `c/n` creates Apple_UNIX_SVR2 partitions.

When creating new partitions, pdisk/fdisk will ask you for the first block of the partition, its length in blocks, and the name of the partition. You can define the first block in one of two ways:
- by its actual location as specified in the 'base' column of the partition map, or
- by the partition number itself, followed by the letter 'p.'

For example, the first block of partition 5 in the map above can be defined as 704 or 5p.

The length in blocks of the partition can also be be defined one of two ways:
- by the number of blocks the partition occupies, or
- the size of the partition in terms of kilobytes, megabytes, and gigabytes followed by the letter k, m, or g, respectively. If you define it in terms of k, m, or g, the numbers must be integers (no decimal points).

If you wanted to create a 2.5 megabyte partition, you could use 5120 as the length in terms of blocks (2048 x 2.5) or you could use 2560k (= 1024 x 2.5). Using 2.5m would not work, as the decimal point will throw off pdisk/fdisk.

One additional note: if you want to enter a partition name that has spaces in it, you need to put quotes (") around it when you enter it.

I think easiest way to learn how to partition is to see for yourself how it is done - with that in mind, I'll go through the steps on splitting the one large Apple_HFS partition at partition 5 into four partitions:

1. MacOS partition, type Apple_HFS, approx. 2 gigabytes
2 .root partition, type Apple_UNIX_SVR2, 60 MB
3. swap partition, type Apple_UNIX_SVR2, 60 MB
4. usr partition, type Apple_UNIX_SVR2, 280 MB

(user input in RED)

```
Command (? For help): d 5
Command (? For help): N
First block: 704  (you can also use 5p instead)
Length in blocks: 4188848  (you can also use 2g instead)
Name of partition: MacOS
Type of partition: Apple_HFS
Command (? For help): p
/dev/hda
   #:                type name                    length   base       ( size )
   1:  Apple_partition_map Apple                     63 @ 1
   2:       Apple_Driver43 Macintosh                 64 @ 64
   3:     Apple_Driver_ATA Macintosh                 64 @ 128
   4:        Apple_Patches Patch Partition          512 @ 192
   5:            Apple_HFS MacOS                4188848 @ 704        (  2.0G)
   6:           Apple_Free Extra                 819200 @ 4189552    (400.0M)

Block size=512, Number of Blocks=5008751
DeviceType=0x0, DeviceId=0x0
Drivers-
1: @ 64 for 20, type=0x1
2: @ 128 for 31, type=0x701

Command (? For help): n
First block: 4189552  (you can also use 6p instead)
Length in blocks: 122880  (you can also use 60m instead)
Name of partition: root
Command (? For help): p
/dev/hda
   #:                type name                    length   base       ( size )
   1:  Apple_partition_map Apple                     63 @ 1
   2:       Apple_Driver43 Macintosh                 64 @ 64
   3:     Apple_Driver_ATA Macintosh                 64 @ 128
   4:        Apple_Patches Patch Partition          512 @ 192
   5:            Apple_HFS MacOS                4188848 @ 704        (  2.0G)
   6:      Apple_UNIX_SVR2 root                  122880 @ 4189552    ( 60.0M)
   7:           Apple_Free Extra                 696320 @ 4312432    (340.0M)

Block size=512, Number of Blocks=5008751
DeviceType=0x0, DeviceId=0x0
Drivers-
1: @ 64 for 20, type=0x1
2: @ 128 for 31, type=0x701

Command (? For help): n
First block: 4312432  (you can also use 7p instead)
Length in blocks: 122880  (you can also use 60m instead)
Name of partition: swap
Command (? For help): p
/dev/hda
   #:                type name                    length   base       ( size )
   1:  Apple_partition_map Apple                     63 @ 1
   2:       Apple_Driver43 Macintosh                 64 @ 64
   3:     Apple_Driver_ATA Macintosh                 64 @ 128
   4:        Apple_Patches Patch Partition          512 @ 192
   5:            Apple_HFS MacOS                4188848 @ 704        (  2.0G)
   6:      Apple_UNIX_SVR2 root                  122880 @ 4189552    ( 60.0M)
   7:      Apple_UNIX_SVR2 swap                  122880 @ 4312432    ( 60.0M)
   8:           Apple_Free Extra                 573440 @ 4435312    (280.0M)

Block size=512, Number of Blocks=5008751
```

```
DeviceType=0x0, DeviceId=0x0
Drivers-
1: @ 64 for 20, type=0x1
2: @ 128 for 31, type=0x701

Command (? For help): n
First block: 4435312  (you can also use 8p instead)
Length in blocks: 573440  (you can also use 280m instead)
Name of partition: usr
Command (? For help): p
/dev/hda
   #:                type name                    length   base       ( size )
   1:  Apple_partition_map Apple                      63 @ 1
   2:       Apple_Driver43 Macintosh                  64 @ 64
   3:     Apple_Driver_ATA Macintosh                  64 @ 128
   4:        Apple_Patches Patch Partition           512 @ 192
   5:            Apple_HFS MacOS                 4188848 @ 704        (  2.0G)
   6:      Apple_UNIX_SVR2 root                   122880 @ 4189552    ( 60.0M)
   7:      Apple_UNIX_SVR2 swap                   122880 @ 4312432    ( 60.0M)
   8:      Apple_UNIX_SVR2 usr                    573440 @ 4435312    (280.0M)

Block size=512, Number of Blocks=5008751
DeviceType=0x0, DeviceId=0x0
Drivers-
1: @ 64 for 20, type=0x1
2: @ 128 for 31, type=0x701

Command (? For help):  w
Writing the map destroys what was there before is that okay? [n/y]: y
The partition table has been altered!

Command (? For help): q
```

Print your partition table to display often, so you know you didn't mess up. If you did mess something up, you can simply delete the partitions that you've made and create new ones. Changes are not permanent until you write the partition. Remember to write your partition table when you're done, otherwise, nothing happens.

Appendix E: Open Firmware Boot Menu

OF Boot Menu by Andi Payn

Andi Payn came up with a cool little Open Firmware program that you can put into the boot-command option in Boot Variables to make it give you an OS boot selector. Here is a copy of Andi's posting about it to the linux-pmac mailing list:

```
-------
Subject:        Open Firmware boot menu!
Date:   Sat, 19 Jul 1997 18:17:58 +0000
From:   andi payn [payn@null.net]
To:     Multiple recipients of list (linux-pmac@arvidsjaur.anu.edu.au)


I know I'm not the only one here with a roommate/kid/whatever who can't seem to figure out how to
properly boot the Mac as a Mac and ends up ruining everything by trying... And I know some people envy
the boot menu that you get with MkLinux (and BeOS).

In the true spirit of Linux-pmac, of course, we want a boot menu that doesn't require loading MacOS.
(If you really wanted to do that, here's what you'd have to do: (1) Write an extension for the MacOS
that displays an OS Chooser dialog. If the user chooses MacOS, unload yourself nicely; if the user
chooses Linux, rewrite the boot variables to boot to Linux, then reboot. (2) Change your Linux startup
so you nvsetenv the boot variables back to boot to MacOS. Or just change your shutdown script so it
changes boot variables first...)

There are two ways to get something to run at startup in OpenFirmware. The first, and probably
superior, is to write an nvramrc file and point OF at it. While I can see where, in BootVariables, I
should be setting things, I haven't been able to make this work. I'll mention the details of what I
tried later, and any help would make me very happy, but for now, let's assume that we can't make that
work. There's still another possibility: The boot command. Some people have had to modify their boot
command to wait for their drive to spin up before booting and so forth. I modified mine to display a
boot menu.

Now, my Forth skills are very rusty, and my OF knowledge is still pretty rudimentary, and I just
slapped this together. Hopefully people will improve on my work.

The major limitation here is that you have a limited amount of space to use. I didn't count it, but
I'm guessing it's 255 characters (it took me a few errors and freezes to realize it was throwing away
the end of the string if it got too long).

What does it look like when I boot my computer? Well, I see a menu that looks like this:

<M>ac
<L>inux
<O>F

Then the computer starts slowly counting down in hex from 100. If I hit L (without the shift key!)
during the countdown, it boots Linux/pmac. If I hit O, it gives me the usual OpenFirmware startup info
(BYE to boot MacOS, BOOT to continue, etc.) If I hit M, or anything else for that matter, or wait for
it to finish counting, it boots MacOS.

Actually, I think you have to hold down the key for a second or so or it won't work.

Anyway, it's not beautiful, but it works. My roommate can figure out how to hit the M key, and if he
can't, he can just wait.

You probably want to write your script in SimpleText and copy it (or just copy mine from this email)
and paste it into boot-variables. Or you'll want to write a brief shell script in emacs or vi that
does it. Either way, remember to keep it all on one line...

So to anyone else who used to enter 1-line program contests in Applesoft Basic, this should be a nice
bit of nostalgia.
```

```
Here's the program, formatted to look pretty (that's a relative term):

" <M>ac" cr ."<L>inux" cr ." <O>F" cr
variable i 100 i !
begin
        key?
        if
                key
                dup ascii o = if throw then
                ascii l = if boot then
                bye
        then
        i @ dup . 1- dup i ! 0<> while
        1000000 begin 1- dup 0<> while repeat drop
repeat
forget i
bye

For those with some Forth knowledge, but not too much, I'll explain. The first line displays the menu.
Then we create our loop index variable, i, and set it to 0x100. We're going to loop through the
"begin..repeat" loop 255 times (someone who can remember how to use a +loop could make this
simpler and save us a few characters we could spend elsewhere) or until someone hits a key.

Each time through the loop, we first check to see if a key's been pressed with the "key?" command.
Unfortunately, in OF I think it returns true only if a key is currently being held down, rather than
if there's a key in the buffer (which is why you have to hold the key down).

If a key's pressed, read it (with the "key" command). Then check it against lowercase o. If it
matches, throw. Since we're not catching, OF will catch it and dump the user into the OF screen.
Probably not the prettiest way to do this, but the only thing I could think of.

Then check to see if it's an L, and if so, boot. Note that your boot-device and boot-file have to be
set up right for this to work (you could use eval to pass parameters to boot, but you'll be wasting
characters you don't have to waste...).

Otherwise, another key's been pressed, so boot MacOS with "bye." If you want to change the logic so
invalid keys are ignored, change the line above to "dup ascii l = if boot then" and change this line
to "ascii m = if bye then" and hopefully that won't push you over 255 characters. If you need to
squeeze a few out, change the first line to:

" <M>ac <L>inux <O>F" cr

Now, if no key's been pressed (or a key's been pressed and it's invalid), we're going to display the
index, decrement it, and check it. If it's still non-zero, the loop keeps going (that's what the
"while" does in this begin..repeat loop).

The next line, the "body" of the loop, just loops 0x1000000 times. That's about 16 million, and should
take your computer somewhere in the vicinity of a second (depending on your computer). You can adjust
this if it's off (actually, it's pretty far off for my computer...). Or, better yet, you could use a
command that's made for waiting. I know there must be one, I just don't remember what it is. As a
guess, I'd say "1000 wait" to wait for 1000ms... Play around with it.

After the outer loop's done, and we haven't quit, we know that the user isn't paying attention. So we
clean up (probably not necessary, if you're looking for characters to save you could kill the "forget
i" line) and boot MacOS ("bye").

Here's the whole thing on one line, ready to paste into BootVariables:

" <M>ac" cr ."<L>inux" cr ." <O>F" cr variable i 100 i ! begin key? if key dup ascii o = if throw then
ascii l = if boot then bye then i @ dup . 1- dup i ! 0<> while 1000000 begin 1- dup 0<> while repeat
drop repeat forget i bye

Enjoy it!
-------
```

It's pretty effective, but there are a few problems I had with his configuration:

```
- there should be a '.' at the beginning of the command (right before " <M>ac") otherwise that text
won't show up in OF
- it should be '." <L>inux"' rather than '."<L>inux"' (note the space between the quote and <L>).
Otherwise, OF barfs.

When booting to linux, the residuals of the menu command after the outer loop gets fed to the second
stage QUIK loader.  Normally, my settings in boot-file get sent to quik, but this isn't happening with
the menu command.  This confuses linux bootup on my StarMax 4160.  (in otherwords, quik gets sent the
option 'forget i bye' instead of the usual '/vmlinux root=/dev/hda5' that I use to boot off the IDE).

I can get around this in two ways:
1.  manually type in '/vmlinux root=/dev/hda5' when quik asks for the bootup image.  This gets to be a
pain.

2.  Fix /etc/quik.conf so that it knows what to do if it sees the 'forget I bye'.  I decided to get
rid of the 'forget i' part on the command - it doesn't seem to affect anything, so now all that gets
sent to the quik loader is 'bye'.  All I had to do was add 'alias = bye' to my quik.conf, and it loads
off of 'bye'.  So now my quik.conf on my IDE root looks like this:

# partition = X is only necessary if root isn't the 1st bootable partition
partition = 5
timeout = 0
default = linux
image = /vmlinux
        label = linux
        root = /dev/hda5
        alias = bye
# switch bye to whatever commands comes 1st after the outer loop of the menu
# command (like 'forget').  I haven't tested it though.

I also got rid of the OF option because the throw wasn't working correctly.  I haven't tried to figure
out why yet.

So now my boot-command looks like this:

." Please select:" cr ."   <m>ac" cr ."   <l>inux" cr ." & hold down the key
for a sec." cr variable i 9 i ! begin key? if key dup ascii l = if boot then
bye then i @ dup . 1- dup i ! 0<> while 700000 begin 1- dup 0<> while repeat
drop repeat bye
```

Finally, Harry Eaton has a variation on the boot menu that he included in his fix for OF on the G3 (ftp://ftp.linuxppc.org/users/harry/fixg3.tgz):

```
." Select" cr ." (m)ac" cr ." (l)inux" cr ." (o)f" cr a 0 do 50000 begin
key? if drop leave then 1- dup 0 <> while repeat drop i . loop key? if key
dup 6d = if bye then 6f = if quit then then bootr
```

Appendix F: Information for BeBox Installation

This page on BeBox installation courtesy of William R Sowerbutts <btg@guru.dircon.co.uk>
http://www.guru.dircon.co.uk/belinux/install.html

LinuxPPC

Linux for the BeBox is based on the LinuxPPC distribution. They have some specific information for installing LinuxPPC on the BeBox here.

Chuck Shotton has written up some installation scripts to make life easier when installing. They're detailed on his site.

A Brief Warning

Note that the **entire** contents (each and every partition) of the hard disk you choose to install the standard LinuxPPC distribution on will be erased; you will require two hard disks if you wish to use both BeOS and LinuxPPC on the same machine. This fact was not well documented on the LinuxPPC web site when I installed it. This is not true if you use the patched kernels detailed on this page, however.

Installing LinuxPPC and BeOS

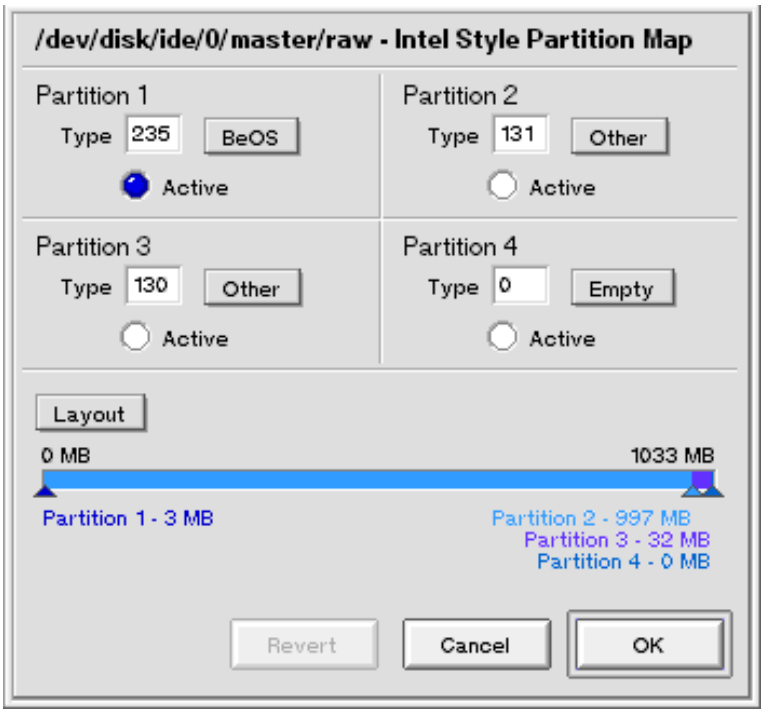
Your mission, should you choose to accept it, is to persuade LinuxPPC and BeOS to share a partition map. The problem is that LinuxPPC and BeOS use different partition maps by default. Since BeOS is the **"native"** operating system of the BeBox, and since we have the means to change the way LinuxPPC behaves but not the way BeOS behaves, we have to change our partitioning scheme to match theirs.

BeOS supports two partition mapping schemes; Intel and Apple style. LinuxPPC supports Intel style partition mapping, so we'll use that one. Since none of the LinuxPPC partitioning software will write out correctly formatted partition maps (yet), you **MUST** do your partitioning under BeOS. You will require a blank hard disk to install on. Our target partition layout is this:

Partition	Type	Contents	Size
1	BeOS (0xEB)	Bootstrap for booting LinuxPPC	At least 3Mb
2	Linux Native (0x83, 131 decimal)	Your root Linux filesystem	Most of the disk
3	Linux Swap (0x82, 130 decimal)	Swap space for Linux	About 32Mb
4	Unpartitioned (0x00)	Unused slot	0Mb

If you want to have BeOS itself on the same hard drive, use the fourth partition. Do not attempt to use the first partition, it won't work. You'll have to install BeOS on the fourth partition and make it the default partition to boot from (there's a Be application for doing that in Preferences somewhere, I'm sure).

Boot the BeOS and load the **Drive Setup** application. Right click on the hard drive you wish to use for Linux, then choose **"Setup"**, **"Partition"**, **"Intel"** . A dialog will pop up. Adjust the settings to match the partition layout given above. For reference, this is how my 1Gb Linux drive looks under Drive Setup:



Format (initialise) the first partition on this disk as a BeOS partition. Give it a name like **"LinuxPPC bootstrap"**. Find a LinuxPPC bootstrap floppy disk, and place it in the floppy drive on your BeBox. I recommend you use the very latest one, which is availablehere. Details on writing it to a floppy are further down this page.

Open the **"Disks"** window, and from the **"File"** menu choose **"Mount Settings"**. Click the **"Mount all disks now"** button. Open your Linux bootstrap disk, and create a folder named **"system"** on it. Open the folder named **"system"** that is on the floppy disk, and copy the file **"kernel"** from the floppy disk to the folder named **"system"** on your newly formatted hard drive. That drive will now load the LinuxPPC bootstrap if you try to boot from it. Now you will need to download some more things (sorry!). The kernel has had to be patched to work with the Be style partition mapping. There are several files available, all by anonymous FTP from ftp.halfast.com in the **/pub/be-linux-devel/** directory. This is a list of the files (just click one to download it).

Filename	Contents	Size
980610-kernel-patches.tgz	This archive file contains the various files I've patched. For the partition mapping, it's **drivers/block/genhd.c**. There are also the results of my fiddling with the interrupt handling code in here.	19Kb
980610-bootstrap.gz	This is the latest version of the bootstrap floppy disk. Uncompress and write this image out to a floppy disk. Leave the floppy disk in the drive and reboot your BeBox to begin loading LinuxPPC.	30Kb
980610-setup.gz	This is a compressed kernel image of the kernel you require to install LinuxPPC. You must have this file to install with a Be-style partition map. Uncompress it before writing it to a floppy disk.	1322Kb
980610-boot-ide.gz	Once you have installed LinuxPPC you must use a second kernel to boot for normal operation. This is a compressed kernel image of the kernel you require if your root disk is on IDE. You must have this file to run Linux with a Be-style partition map. Uncompress it before writing it to a floppy disk.	478Kb
980610-boot-scsi.gz	Once you have installed LinuxPPC you must use a second kernel to boot for normal operation. This is a compressed kernel image of the kernel you require if your root disk is on SCSI. You must have this file to run Linux with a Be-style partition map. Uncompress it before writing it to a floppy disk.	478Kb

Performing the installation

Download the latest bootstrap from the above list. Uncompress and write it out to a formatted 1.44Mb 3.5" floppy disk with a command like this:

```
inferno:~/belinux$ gzip -dc 980610-bootstrap.gz > /dev/fd0
```

Label that floppy disk **Bootstrap**. You will need this disk to prepare the hard drive as a bootstrap (see above). Insert a new one and repeat the process for the setup disk. Change the disk again, and then write out your boot disk - make sure you choose the correct disk, depending on whether the disk you are installing on is IDE or SCSI. Both disks include drivers for IDE and SCSI, but one disk expects to find the root filesystem on the first IDE disk, whereas the other expects to find it on the first SCSI disk. You cannot presently install on disks other than the first SCSI disk (well you can, but you'd need to patch the kernel first).

Place the **"Setup"** disk in your floppy drive, and reboot the BeBox. Hold down the shift key as soon as the Be logo appears (or even from before it appears, if you can). You should get a list of the BeOS filesystems in your system. One of them should be the bootstrap you prepared earlier. If you cannot see it, choose the bottom option, which will cause the system to rescan the SCSI and IDE busses looking for disks. If you still cannot see it after a few seconds, something has gone wrong. Did you copy the bootstrap code to **"system/kernel"** on that disk, as described above? If you still cannot get it to work, something is seriously wrong.Contact me and tell me what you've done.

When you've found the correct disk to boot from, highlight it and press return. You should see the video card change to text mode and the words **"PowerPC Linux for BeBox bootstrap"** followed by a massive wadge of status and debug information. If you just get a black screen, reboot and try again. If you get no change at all, you have an unsupported video card. Sorry about that. We're actively working to add support for more video cards, even as you read this.

With the **"Setup"** floppy in your floppy drive, hit return. It's okay to boot with this floppy in the drive, the boot ROM won't try to load an operating system from it ;)

The system will load the kernel from the floppy disk. As it loads, the two rows of lights on either side of the BeBox will **"fill up"** . After the kernel has loaded, the screen will clear again and you'll be confronted with the**"Linux/PowerPC Load:"** prompt. Just hit return at this point, and wait as the installation kernel and RAM disk (slowly) decompresses.

Finally, the screen will clear once more and Linux will boot into single user mode from the RAM disk built into the setup kernel image. You should get a**"#"** prompt. During the **"Partition check"** you should see **"{BeFix}"** printed for the disk you plan to install on (either sda or hda). If you don't, abandon your installation now and contact me!

Type **"crdisk-net"** and hit return to begin the installation. Choose **"Diskettes"** as your installation method. Enter **"/dev/fd1"** as the device name for your floppy. Make sure you enter the correct device name to install onto - use

"/dev/sda" for the first SCSI disk, and **"/dev/hda"** for the first IDE disk. Name **"dev/hda2"** or **"dev/sda2"** as the root partition, and **"dev/hda3"** or **"dev/sda3"** as the swap partition. When it asks you for the options for **cfdisk** , erase the **"-z"** from the command line! Do **NOT** repartition your hard drive when it tells you to! It will start cfdisk for you, you must quit it without allowing it to alter your partition table. To do this, move the cursor over to **"Quit"** using the right arrow key, then press the return key. You will need to download, decompress, and write some files out to floppy disk to complete the installation. These are all found on the LinuxPPC FTP site.

Once you've completed the installation you'll get the **"#"** prompt again. Reboot the BeBox and start the bootstrap as before. Instead of using the **"Setup"** disk to load the kernel, however, use one of the **"boot-ide"** or **"boot-scsi"** disks. You may now install RedHat PowerPC RPMs to get a working LinuxPPC installation.

Below are some installation scripts for the BeBox, courtesy of Chuck Schotten. These can also be found at his website at http://www.biap.com/Linux/be.html

Be Linux Install Scripts

This is a quick and dirty page describing 3 scripts for use in installing the LinuxPPC 98 CD distribution on BeBox hardware. That CD installer will not work with the BeBox, hence the need for these scripts.

This page assumes you have a basic working knowledge of Unix-like operating systems and can create directories, edit files, etc. The scripts seem to work, but sometimes kernel crashes do occur during execution. If this happens, you should modify the scripts to resume where you left off, rather than rerunning the complete script.

These scripts are based on information found at Robert Currey's site (http://www.halcyon.com/curreyr/BeLinux.html). What is important is that they attempt to document the minimal packages that need to be installed and the basic order in which they should be installed. If the script execution fails, you may be better off running the individual commands by hand.

Finally, you should start with the 2.0.30 kernel available from ftp.linuxppc.org. These scripts should be run after performing a floppy-based install as described at that site. A network-based install that uses the TAPE_FILE3.gz file system will probably not allow X Windows to be installed later.

Install script 1

This script creates a mount point for the CD, mounts it, and installs the basic libraries and utility functions necessary to continue.

```
Mkdir /mnt/cd
mount -o ro /dev/sr0 /mnt/cd
cd /mnt/cd/RedHat/RPMS
rpm -ihv --nodeps glibc-0*
rpm -ihv ldconf*
rpm -ihv --nodeps glibc-info*
rpm -ihv --nodeps glibc-static*
rpm -ihv --nodeps libc-5*
rpm -ihv --nodeps libc-static*
rpm -ihv --nodeps libg++-2*
rpm -ihv --nodeps libgr-2*
rpm -ihv --nodeps libgr-progs*
```

```
rpm -ihv --nodeps libjpeg-*
rpm -ihv --nodeps libpng-0*
rpm -ihv --nodeps libtermcap-2*
rpm -ihv --nodeps zlib-1*
rpm -ihv --nodeps pwdb-0*
rpm -ihv --nodeps perl-*
rpm -ihv --nodeps ncurses-*
rpm -ihv --nodeps nvi-*
echo ""
echo "Now edit the /etc/ld.so.conf file and add the following lines:"
echo "/lib"
echo "/usr/lib"
echo "/usr/local/lib"
echo "/usr/X11/lib   -- only necessary if you plan on installing X"
echo ""
echo "Next, run ldconfig as follows:"
echo "/sbin/ldconfig -v -f /etc/ld.so.conf"
echo ""
echo "Finally, reboot the BeBox  (shutdown -r now) and continue with the"
echo "second install script, be2."
Echo ""
```

Install script 2

This script installs additional tools, networking functions, and documentation.

```
Mount -o ro /dev/sr0 /mnt/cd
cd /mnt/cd/RedHat/RPMS
rpm -ihv --nodeps binu*
rpm -ihv --nodeps less-*
rpm -ihv --nodeps setup-*
rpm -ihv --nodeps etcskel-*
rpm -ihv --nodeps rootfiles-*
rpm -ihv --nodeps bash-*
rpm -ihv --nodeps diffutils-*
rpm -ihv --nodeps e2fsprogs-*
rpm -ihv --nodeps textutils-*
rpm -ihv --nodeps findutils-*
rpm -ihv --nodeps cpio-*
rpm -ihv --nodeps fileutils-*
rpm -ihv --nodeps gawk-3*
rpm -ihv --nodeps sed-2*
rpm -ihv --nodeps cracklib-*
rpm -ihv --nodeps pam-0*
rpm -ihv --nodeps sh-utils-*
rpm -ihv --nodeps util-linux-*
rpm -ihv --nodeps rpm-2*
rpm -ihv --nodeps rpm-*
rpm -ihv --nodeps passwd-0*
rpm -ihv --nodeps termcap-9*
rpm -ihv --nodeps groff-1*
rpm -ihv --nodeps man-1*
rpm -ihv --nodeps man-pages*
rpm -ihv --nodeps SysVinit-*
rpm -ihv --nodeps slang-*
rpm -ihv --nodeps newt-*
rpm -ihv --nodeps tcp_wrappers*
rpm -ihv --nodeps net-tools-*
```

```
rpm -ihv --nodeps netkit*
rpm -ihv --nodeps MAKEDEV*
rpm -ihv --nodeps sysklogd*
rpm -ihv --nodeps telnet*
rpm -ihv --nodeps wu*
rpm -ihv --nodeps lynx*
rpm -ihv --nodeps adduser*
/sbin/ldconfig -v -f /etc/ld.so.conf
echo ""
echo "Reboot the BeBox to complete the basic installation. (shutdown -r now)"
```

X Windows Install script

This script installs the basic X Windows client functionality. You likely won't be able to run a X server on the local BeBox console with these files, but the packages below will make it possible to talk to the BeBox via X with a server running on a separate platform.

```
Mount -o ro /dev/sr0 /mnt/cd
cd /mnt/cd/RedHat/RPMS
rpm -ihv --nodeps --force xinitrc-*
rpm -ihv --nodeps --force X11R6.3-0*
rpm -ihv --nodeps --force X11R6.3-100dpi-fonts*
rpm -ihv --nodeps --force X11R6.3-75dpi-fonts*
rpm -ihv --nodeps --force X11R6.3-devel*
rpm -ihv --nodeps --force X11R6.3-fonts*
rpm -ihv --nodeps --force X11R6.3-libs*
rpm -ihv --nodeps --force X11R6.3-man*
/sbin/ldconfig -v -f /etc/ld.so.conf
```

Additional Info

If you have additional information to contribute, suggestions, or changes, please send e-mail to cshotton@biap.com.

Appendix G: Information for CHRP Installation

Installation information on CHRP systems is courtesy of Geert Uytterhoeven.
This information can also be found at his website at http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/

This section discusses the basic installation of Linux/PPC on the CHRP LongTrail.

Since the floppy drive isn't working yet under Linux, the easiest installation method is using a root file system on NFS. Make sure you have a second computer available that can act as a RARP/BOOTP and NFS server. Alternatively you can format a hard disk on another machine (msdos partitioning), install the root filesystem and move the disk to your CHRP machine.

Files

You can download the following files from this page:

Linux kernels
There are two kernels available, one with support for ADB keyboards and mice, and another with support for a PS/2 keyboard and mouse (you can't have both ADB and PS/2 support in the same kernel (yet)). Both kernels support DEC21041 Ethernet and ATI video boards. If you have a different video board, it will still work, as long as Open Firmware knows how to initialize the board.

ZImage.adb (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/zImage.adb)
Kernel image with support for ADB keyboards and mice
.config.adb (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/.config.adb)
Kernel configuration for the above kernel image
System.map.adb (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/System.map.adb)
System.map for the above kernel image
zImage.ps2 (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/zImage.ps2)
Kernel image with support for a PS/2 keyboard and mouse
.config.ps2 (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/.config.ps2)
Kernel configuration for the above kernel image
System.map.ps2 (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/System.map.ps2)
System.map for the above kernel image

Root filesystem
chrproot.tar.gz (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/chrproot.tar.gz)
This is a modified version of TAPE_FILE3.gz, which supports the new console device minor.

Extract this archive on your NFS server.

Kernel source patches
chrp-2.1.72-19971223.diff.gz (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/chrp-2.1.72-19971223.diff.gz) brings your 2.1.72 kernel source tree to the same level as the `bleeding edge' source tree at vger.rutgers.edu (dated December 22, 1997).

Booting Linux

Open Firmware booting is nice: just copy the kernel to a msdos formatted floppy and type

```
boot a:<kernel-name> root=/dev/nfs
```

The kernel has support for both RARP and BOOTP to find out it's IP address (and hostname). It will mount (using NFS)

```
/tftpboot/<ip-address>
```

or

```
/tftpboot/<hostname>
```

(if BOOTP supplied a hostname) as its root file system. If you want to disable RARP and BOOTP, add `ip=off' to the boot
command.

My /etc/bootptab looks like

```
callisto:\
        :hn:ht=ethernet:vm=rfc1048:\
        :ha=0080c85af85b:
```

`callisto` is the name of my CHRP machine, `0080c85af85b` is the hardware address of my Ethernet board.

If Open Firmware doesn't support your video board, you can still boot Linux using a serial terminal, connected to one of the 9 pin D-SUB serial ports, as the console, by adding

```
console=ttyS0
```

or

```
console=ttyS1
```

to the boot command.

The first things the boot loader says are:

```
Boot device: <boot-device> File and args: <file-and-args>
chrpboot starting
gunzipping… done
start address = 0x9000100c
copying OF device tree… done
instantiating rtas… done
```

The copy process takes about 10-15 seconds.

After this you should see the well known Linux penguin logo and the kernel boot messages, and you'll be thrown into a
single user shell. Then you can partition your hard disk and copy the root file system to it.

Good luck!

Miscellaneous

- If you don't specify a root file system, the default is `/dev/sda2`.
- Of course you can put kernels on hard disk too. Just create a msdos formatted partition (e.g. on `/dev/sda1`) and
  copy your kernels to it. I use
  `boot scsi/disk@6,0:1,zImage ip=off`
  to boot from the file `zImage` on the first partition of the SCSI disk at unit 6 LUN 0.
- IDE disks are not yet supported. Besides, IDE sucks! SCSI lives!

Appendix H: Information for PReP Installation

Installation information on PreP systems is courtesy of Cort Dougan.
This information can also be found at his website at http://linuxppc.cs.nmt.edu/getit.html

---

How to get the system

What you need to get for the PReP systems. See Paul Mackerras' info for what you need for the PowerMac systems using this kernel. The current pmac archive is here (ftp://samba.anu.edu.au/pub/linux-pmac/). We are slowly merging to one binary for pmac/prep/chrp.

If you have problems: email cort@cs.nmt.edu with questions

Files you need for the RedHat based install method:

RedHat-style installer

> RedHat-style installer (ftp://linuxppc.cs.nmt.edu/pub/linuxppc/install/installer.prep)-- Just dd or raw-write this image onto a floppy and boot it

RedHat-style packages

> RedHat-style PPC packages (ftp://linuxppc.cs.nmt.edu/pub/linuxppc/install/)-- You can install these with the install program by allowing it to ftp them during the install or you can retrieve them now.

Boot image -- for after you have installed your system

> Zimage (ftp://linuxppc.cs.nmt.edu/pub/linuxppc/kernel-images/zImage)

Thank you for supporting the LinuxPPC Project.